

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2000-311104

(43)Date of publication of application : 07.11.2000

(51)Int.Cl. G06F 12/00
G06F 3/06

(21)Application number : 2000-038062 (71)Applicant : SONY CORP

(22)Date of filing : 09.02.2000 (72)Inventor : IIDA KENICHI
YAMADA EIICHI
OBAYASHI HIDEJI

(30)Priority

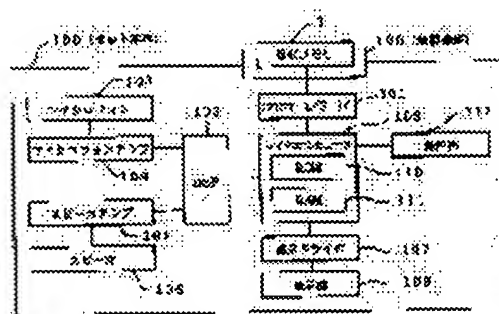
Priority number : 11051270 Priority date : 26.02.1999 Priority country : JP

(54) METHOD FOR RECORDING, METHOD FOR MANAGEMENT, AND RECORDING DEVICE

(57)Abstract:

PROBLEM TO BE SOLVED: To carry out recording corresponding to high-bit-rate stream data by recording generated link information and a recording start position in a management area after the recording of main information to a main information storage area and erasing a temporary file name.

SOLUTION: When a plate type memory 1 is loaded in a set main body 100, a microcomputer 109 reads in a data part as main information from a logical-physical address conversion table in the plate type memory 1 through a host interface IC 101 and records it in the main information storage area of a RAM 111. After the recording, only cluster numbers are recorded in the RAM 111 each time the stream data is recorded in the plate type memory 1 so that the recording order can be discriminated; when the links of the recorded clusters are processed, link information is generated according to the cluster numbers, the link information and recording start position are recorded in the management



area of the RAM 111 and the temporary file name is erased.

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2000-311104

(P2000-311104A)

(43) 公開日 平成12年11月7日 (2000.11.7)

(51) Int.Cl. ⁷	識別記号	F I	データ形式 (参考)
G 0 6 F 12/00	5 2 0	G 0 6 F 12/00	5 2 0 J
	5 1 4		5 1 4 A
3/06	3 0 1	3/06	3 0 1 K

審査請求 未請求 請求項の数21 O L (全 29 頁)

(21) 出願番号 特願2000-38062(P2000-38062)

(22) 出願日 平成12年2月9日 (2000.2.9)

(31) 優先権主張番号 特願平11-51270

(32) 優先日 平成11年2月26日 (1999.2.26)

(33) 優先権主張国 日本 (J P)

(71) 出願人 000002185
ソニー株式会社
東京都品川区北品川6丁目7番35号

(72) 発明者 飯田 健一
東京都品川区北品川6丁目7番35号 ソニー株式会社内

(72) 発明者 山田 榮一
東京都品川区北品川6丁目7番35号 ソニー株式会社内

(72) 発明者 大林 秀次
東京都品川区北品川6丁目7番35号 ソニー株式会社内

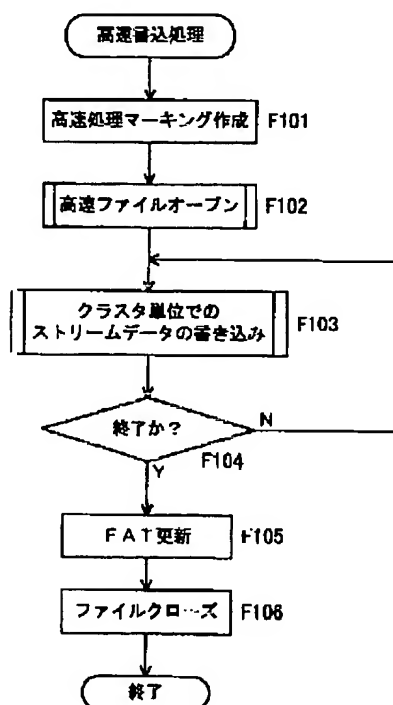
(74) 代理人 100086841
弁理士 脇 篤夫

(54) 【発明の名称】 記録方法、管理方法、及び記録装置

(57) 【要約】

【課題】 高ビットレートのストリームデータに対応して書込可能とする。

【解決手段】 ストリームデータ等の主情報を連続して記録している期間には、ディレクトリ情報やリンク情報等の管理情報の更新は行なわないようにすることで、主情報の記録に関して、単位時間当たりの書込可能なデータ量を大幅に増大させる。



【特許請求の範囲】

【請求項1】 ブロック化された単位領域を有し、時間的に連続して入力される主情報を、離散的なブロック又は連続的なブロックに記録していくことができる主情報記録領域と、

上記主情報に対する上記主情報記録領域上の記録開始位置と、上記主情報が記録された各ブロックを論理的に結合させるリンク情報と、上記主情報を管理するディレクトリ情報と、から成る管理情報を記録する管理領域とから構成される記録媒体に、主情報を記録する記録方法として、

新規な主情報を記録する際に上記ディレクトリ情報として暫定ファイル名を記録する暫定ファイル名記録手順と、

上記主情報記録領域に存在する記録可能なブロックを所定の順序で検索する検索手順と、

上記検索手順において検知した記録可能なブロックに上記主情報を順次記録する主情報記録手順と、

上記主情報記録手順による主情報の上記主情報記録領域への記録が終了した後に、上記リンク情報を生成して、生成されたリンク情報と記録開始位置を上記管理領域に記録する管理情報記録手順と、

上記暫定ファイル名を消去する暫定ファイル名消去手順と、
が行われることを特徴とする記録方法。

【請求項2】 上記検索手順での記録可能なブロックの検索は、ブロックに付与された番号を昇順に検索していくことを特徴とする請求項1に記載の記録方法。

【請求項3】 上記主情報記録手順での、上記記録可能なブロックへの主情報の記録は、上記検索手順で検索された記録可能なブロックの番号を記録媒体の物理アドレスに変換してから行うことを特徴とする請求項1に記載の記録方法。

【請求項4】 上記主情報記録手順では、上記記録可能なブロックへの主情報の記録終了前に、次に記録可能なブロックの内容を消去するようにしたことを特徴とする請求項1に記載の記録方法。

【請求項5】 上記管理情報記録手順では、上記管理情報で記録可能とされているブロックのうちで主情報が既に記録されたブロックを検索して、そのブロックの番号を順に結合させることで上記リンク情報を作成することを特徴とする請求項1に記載の記録方法。

【請求項6】 上記リンク情報の作成の際には、上記管理情報で記録可能とされているブロックの番号を所定の順に検索して行き、上記主情報が記録されていないブロックを検出した際の直前のブロックを、上記主情報が最後に記録されたブロックと判別してリンク情報を作成することを特徴とする請求項5に記載の記録方法。

【請求項7】 上記主情報記録手順では、最後に主情報を記録したブロックに付与された番号を記憶しておき、

上記管理情報記録手順では、上記主情報記録手順で記憶した番号を最後に主情報の記録されたブロックと判別してリンク情報を作成することを特徴とする請求項5に記載の記録方法。

【請求項8】 上記暫定ファイル名とともに、新たな主情報を最初に記録するブロックに付与された番号を記憶しておき、

上記管理情報記録手順では、上記暫定ファイル名とともに記憶した番号を主情報の記録された最初のブロックと判別してリンク情報を作成することを特徴とする請求項5に記載の記録方法。

【請求項9】 上記暫定ファイル名消去手順では、上記暫定ファイル名を消去するとともに、記録された主情報を示すファイル名を上記管理領域に記録することを特徴とする請求項1に記載の記録方法。

【請求項10】 ブロック化された単位領域を有し、時間的に連続して入力される主情報を、離散的なブロック又は連続的なブロックに記録していくことができる主情報記録領域と、

上記主情報に対する上記主情報記録領域上の記録開始位置と、上記主情報が記録された各ブロックを論理的に結合させるリンク情報と、上記主情報を管理するディレクトリ情報と、から成る管理情報を記録する管理領域とから構成される記録媒体を管理する管理方法として、

上記ディレクトリ情報に暫定ファイル名が書き込まれているか否かを判定する暫定ファイル名判定手順と、

上記判定手順によって暫定ファイル名が書き込まれていると判定された場合に、上記管理情報に基づいて記録可能とされるブロックを所定の順序で検索する検索手順と、

上記検索されたブロックに主情報が記録されているか否かを判定する記録判定手順と、

上記記録判定手順において最初に主情報が記録されたブロックであると判定されたブロックを記録開始位置とし、さらに上記検索手順の検索順序で主情報が記録されていると判定されたブロックが論理的に結合されるようにリンク情報を作成するリンク情報作成手順と、

上記記録開始位置と上記リンク情報を、上記管理情報として上記管理領域に記録する管理情報記録手順と、
上記暫定ファイル名を消去する暫定ファイル名消去手順と、

が行われることを特徴とする管理方法。

【請求項11】 上記検索手順での所定順序の検索は、ブロックに付与された番号の昇順に行うことを特徴とする請求項10に記載の管理方法。

【請求項12】 上記リンク情報作成手順では、上記管理情報で記録可能とされているブロックの番号の所定の順の検索において、上記主情報が記録されていないブロックを検出した際の直前のブロックを、上記主情報が最後に記録されたブロックと判別してリンク情報を作成す

ることを特徴とする請求項10に記載の管理方法。

【請求項13】 ブロック化された単位領域を有し、時間的に連続して入力される主情報を、離散的なブロック又は連続的なブロックに記録していくことができる主情報記録領域と、

上記主情報に対する上記主情報記録領域上の記録開始位置と、上記主情報が記録された各ブロックを論理的に結合させるリンク情報と、上記主情報を管理するディレクトリ情報と、から成る管理情報を記録する管理領域とから構成される記録媒体に対して、主情報を記録する記録装置として、

上記ディレクトリ情報と上記主情報と上記管理情報の各々を、上記記録媒体の所定の領域に記録または消去することができる記録消去手段と、

上記記録媒体の記録可能なブロックを所定の順番で検索する検索手段と、

リンク情報を生成するリンク情報生成手段と、

新規な主情報を記録する際に暫定ファイル名を上記ディレクトリ情報として上記管理領域に記録させ、また連続して入力される主情報を上記検索手段により検索された記録可能なブロックに順次記録させ、また上記主情報の記録が終了した後に上記リンク情報生成手段によって生成されたリンク情報と記録開始位置とを上記管理領域に記録させ、また上記暫定ファイル名を消去させるように、上記記録消去手段を制御する制御手段と、を備えていることを特徴とする記録装置。

【請求項14】 上記検索手段は、ブロックに付与された番号の昇順に検索を行うことを特徴とする請求項13に記載の記録装置。

【請求項15】 上記検索手段で検索された記録可能なブロックの番号を記録媒体の物理アドレスに変換する変換手段を更に備え、

上記記録消去手段による記録可能なブロックへの主情報の記録は、上記物理アドレスに基づいて行われることを特徴とする請求項13に記載の記録装置。

【請求項16】 上記制御手段は、上記記録可能なブロックへの主情報の記録が終了する前に、次に記録可能なブロックの記録内容を上記記録消去手段によって消去させることを特徴とする請求項13に記載の記録装置。

【請求項17】 上記リンク情報生成手段は、上記検索手段によって検索された記録可能なブロックのうち主情報が既に書き込まれたブロックの番号を所定の順に結合させて上記リンク情報を生成することを特徴とする請求項13に記載の記録装置。

【請求項18】 上記検索手段によって上記所定の順序で検索された記録可能なブロックのうちで、主情報が記録されていた最後のブロックを、上記主情報の連続した記録における最後のブロックと判断することを特徴とする請求項13に記載の記録装置。

【請求項19】 上記制御手段は、上記記録消去手段に

より上記暫定ファイル名を消去させるとともに新たに記録した主情報を示すファイル名を上記管理領域に記録させることを特徴とする請求項13に記載の記録装置。

【請求項20】 上記記録媒体が着脱可能とされていることを特徴とする請求項13に記載の記録装置。

【請求項21】 上記記録媒体は、フラッシュメモリで構成されることを特徴とする請求項13に記載の記録装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は各種データファイルを記録できる記録媒体に対するデータファイルの記録方法、管理方法、及び記録装置に関するものである。

【0002】

【従来の技術】近年、例えばフラッシュメモリなどの固体記憶素子を搭載した小型の記録媒体を形成し、専用のドライブ装置や、或いはドライブ装置をオーディオ/ビデオ機器、情報機器などに内蔵して、コンピュータデータ、静止画像データ、動画データ、音楽や音声のオーディオデータなどを記憶できるようにするものが開発されている。

【0003】

【発明が解決しようとする課題】ところで、このような固体記憶素子を利用した記録システムにおいては、動画データ、オーディオデータなど、内容的に時間的連続性があり、記録データとしては時間的に連続して供給されてくるデータ（以下、ストリームデータという）を記録する場合には、単位時間当たりの記録可能データ量とストリームデータのビットレートの関係で、適切に記録ができない場合がある。換言すれば、時間的に連続して供給されてくるストリームデータのビットレート（単位時間当たりのデータ量）が記録可能なデータ量としてのビットレートを超える場合は、ストリームデータがオーバーフローしてしまうことになる。そのため、記録可能なストリームデータとしてのビットレートが制限されることになる。つまりビットレートが所定値以上高いストリームデータはリアルタイムには書込ができない。

【0004】ここで、フラッシュメモリを用いる記録媒体に対して、FAT (File Allocation Table) により書込データのアドレスを管理するとともに、アドレス変換テーブルを用いるシステムについてのデータ書込処理について図32に示す。FAT及びアドレス変換テーブルについて詳しくは後述するが、FATとは所定データ単位（後述するクラスタ）毎にデータを書き込んだ際に、そのアドレスをリンクさせることで1つのファイルとしての各クラスタのつながりを管理する情報である。またアドレス変換テーブルは、FATで用いる論理アドレスを実際の記録媒体上の実アドレスである物理アドレスに変換するための情報である。即ち或る1つのファイルデータは、記録媒体上で通常複数のクラスタに渡って

記録されるが、FATではそのクラスタのつながりを記録することで、1つのデータファイルとしての記録部分（各クラスタのアドレス及び順序）が管理される。またFATでは論理アドレスとしてデータに対応するアドレスで処理を行うが、これがアドレス変換テーブルに基づいて物理アドレスに変換されて記録媒体に対する記録再生アクセスが行われることになる。

【0005】ストリームデータの記録が行われる場合には、記録装置のコントローラは、図32に示すようにまずステップF301としてファイルオープンを行なって記録動作を開始する。そして供給されてくるストリームデータとして、1クラスタ分のデータが取り込まれる毎に、ステップF302でそのクラスタ分のデータの記録媒体への書込を行い、ステップF303、F304でクラスタの書込に伴ったアドレス変換テーブル及びFATの更新を行う。つまり記録媒体上のFAT及びアドレス変換テーブルを書き換える。

【0006】これらの更新処理が必要なのは、クラスタデータの書き込みには、FAT上の未使用の論理アドレスを割り当てて使用し、記録媒体上ではこの論理アドレスに対応した物理アドレスが使用されるためである。つまり、1つのクラスタの記録に応じてFAT上のある未使用領域が割り当てられ、その論理アドレスに対応させて未使用の物理アドレスをアドレス変換テーブル上で割り当てるためである。

【0007】1クラスタ分のストリームデータが取り込まれる毎にこのステップF302～F304を繰り返していき、記録データとして供給された全ストリームデータの記録が完了したら、ステップF305からF306に進み、ファイルクローズを行って処理を終える。

【0008】このような処理が通常行われるわけであるが、この場合、1クラスタ分のデータを書き込む毎に、FATの更新データの書込及びアドレス変換テーブルの更新データの書込が必要になる。つまりストリームデータ量：1に対して管理情報更新データ量：2の割合で書込が必要になるため、ストリームデータの書込可能量は単位時間当たりの書込可能データ量の約1/3となっていた。

【0009】記録媒体に書込可能な単位時間当たりのデータ量はシステムのハードウェアのスペックにより決まってしまうため、上述したようにこれを越えるようなビットレートのストリームデータに対応不能となることは仕方ないが、上記処理の事情により、書込可能な単位時間当たりのデータ量の約1/3など、ハードウェアのスペックによる上限よりも遙かに低い上限が生じてしまうことは好ましいものではない。このようなことから、或る1つのシステムとして、ハードウェア的には可能であるが、ソフトウェア的な事情により高ビットレートのストリームデータの書込に対応できないというようなことをできるだけ解消したいという要望がある。言い換え

ば、可能な限りハードウェアのスペックにより決まる上限に近いビットレートのストリームデータまで記録可能とさせたい。

【0010】

【課題を解決するための手段】本発明はこのような事情に応じて、ブロック化された単位領域を有し、時間的に連続して入力される主情報を、離散的なブロック又は連続的なブロックに記録していくことができる主情報記録領域と、上記主情報に対する上記主情報記録領域上の記録開始位置と、上記主情報が記録された各ブロックを論理的に結合させるリンク情報と、上記主情報を管理するディレクトリ情報と、から成る管理情報を記録する管理領域とから構成される記録媒体に対して、効率的な記録処理を実行することで、より高ビットレートのストリームデータにも対応して記録を実行できるようにすることを目的とする。

【0011】このため上記記録媒体に主情報を記録する本発明の記録方法は、新規な主情報を記録する際に上記ディレクトリ情報として暫定ファイル名を記録する暫定ファイル名記録手順と、上記主情報記録領域に存在する記録可能なブロックを所定の順序で検索する検索手順と、上記検索手順において検知した記録可能なブロックに上記主情報を順次記録する主情報記録手順と、上記主情報記録手順により、主情報の上記主情報記録領域への記録が終了した後に、上記リンク情報を生成して、生成されたリンク情報と記録開始位置を上記管理領域に記録する管理情報記録手順と、上記暫定ファイル名を消去する暫定ファイル名消去手順と、が行われるようにする。

【0012】また上記記録媒体に対する本発明の管理方法は、上記ディレクトリ情報に暫定ファイル名が書き込まれているか否かを判定する暫定ファイル名判定手順と、上記判定手順によって暫定ファイル名が書き込まれていると判定された場合に、上記管理情報に基づいて記録可能とされるブロックを所定の順序で検索する検索手順と、上記検索されたブロックに主情報が記録されているか否かを判定する記録判定手順と、上記記録判定手順において最初に主情報が記録されたブロックであると判定されたブロックを記録開始位置とし、さらに上記検索手順の検索順序で主情報が記録されていると判定されたブロックが論理的に結合されるようにリンク情報を作成するリンク情報作成手順と、上記記録開始位置と上記リンク情報を、上記管理情報として上記管理領域に記録する管理情報記録手順と、上記暫定ファイル名を消去する暫定ファイル名消去手順と、が行われるようにする。

【0013】また上記記録媒体に対する本発明の記録装置は、上記ディレクトリ情報と上記主情報と上記管理情報の各々を、上記記録媒体の所定の領域に記録しまたは消去することができる記録消去手段と、上記記録媒体の記録可能なブロックを所定の順番で検索する検索手段と、リンク情報を生成するリンク情報生成手段と、新規

な主情報を記録する際に暫定ファイル名を上記ディレクトリ情報として上記管理領域に記録させ、また連続して入力される主情報を上記検索手段により検索された記録可能なブロックに順次記録させ、また上記主情報の記録が終了した後に上記リンク情報生成手段によって生成されたリンク情報と記録開始位置とを上記管理領域に記録させ、また上記暫定ファイル名を消去させるように、上記記録消去手段を制御する制御手段と、を備えているようにする。

【0014】即ち本発明では、ストリームデータ等の主情報を記録している期間には、管理情報やアドレス変換情報の更新は行わないようにする（別の時点で行う）。これにより主情報の記録に関して、単位時間当たりの書込可能データ量を大幅に増大させ、より高ビットレートのストリームデータにも対応できるようにする。

【0015】

【発明の実施の形態】以下、本発明の実施の形態について説明する。なお、この実施の形態は、記録媒体として板状の外形状を有する板状メモリを挙げ、これに対してデータの記録再生を行うことのできるドライブ装置及びそのドライブ装置による記録方法、管理方法とする。説明は次の順序で行う。

1. 板状メモリの外形形状
2. 板状メモリのフォーマット
 - 2-1. メモリファイルシステム処理階層
 - 2-2. 物理的データ構造
 - 2-3. 物理アドレス及び論理アドレスの概念
 - 2-4. 論理-物理アドレス変換テーブル
3. ドライブ装置の構成
4. FAT構造
5. ストリームデータの書込処理
6. 復活処理

【0016】1. 板状メモリの外形形状

まず図1により、本例の記録媒体である、板状メモリ1の外形形状について説明する。板状メモリ1は、例えば図1に示すような板状の筐体内部に例えば所定容量のメモリ素子を備える。本例としては、このメモリ素子としてフラッシュメモリ（Flash Memory）が用いられるものである。図1に平面図、正面図、側面図、底面図として示す筐体は例えばプラスチックモールドにより形成され、サイズの具体例としては、図に示す幅W11、W12、W13のそれぞれが、W11=60mm、W12=20mm、W13=2.8mmとなる。

【0017】筐体の正面下部から底面側にかけて例えば9個の電極を持つ端子部2が形成されており、この端子部2から、内部のメモリ素子に対する読出又は書込動作が行われる。筐体の平面方向の左上部は切欠部3とされる。この切欠部3は、この板状メモリ1を、例えばドライブ装置本体側の着脱機構へ装填する際に挿入方向を誤ることを防止するためのものとなる。また筐体底面

側には使用性の向上のため滑り止めを目的とした凹凸部4が形成されている。さらに底面側には、記憶内容の誤消去を防止する目的のスライドスイッチ5が形成されている。

【0018】2. 板状メモリのフォーマット

2-1. メモリファイルシステム処理階層

続いて、板状メモリ1を記録媒体とするシステムにおけるフォーマットについて説明していく。図2は、板状メモリ1を記録媒体とするシステムのファイルシステム処理階層を示すものである。この図に示すように、ファイルシステム処理階層としては、アプリケーション処理層の下に、順次、ファイル管理処理層、論理アドレス層、物理アドレス層、フラッシュメモリアクセスがおかれる。この階層では、ファイル管理処理層がいわゆるFAT（File Allocation Table）となる。また、この図から分かるように、本例のファイルシステムでは論理アドレス及び物理アドレスという概念が導入されているが、これについては後述する。

【0019】2-2. 物理的データ構造

図3には、板状メモリ1内の記憶素子である、フラッシュメモリの物理的データ構造が示されている。フラッシュメモリとしての記憶領域は、セグメントという固定長のデータ単位が大元となる。このセグメントは、1セグメントあたり4MB（メガバイト）或いは8MBとして規定されるサイズであり、1つのフラッシュメモリ内におけるセグメント数は、そのフラッシュメモリの容量に依存して異なってくる。

【0020】そして、この1セグメントを図3（a）に示すように、ブロックという固定長のデータ単位として8KB（キロバイト）又は16KBにより区切るようにされる。原則として、1セグメントは512ブロックに区切られることから、図3（a）に示すブロックnについては、 $n=511$ とされることになる。但し、フラッシュメモリでは、書き込み不可な損傷エリアであるディフェクトエリアとしてのブロック数が所定数の範囲で許可されているため、データ書き込みが有効とされる実質的なブロック数を対象とすれば、上記nは511よりも少なくなる。

【0021】図3（a）に示すようにして形成されるブロック0～nのうち、先頭の2つのブロック0、1はブートブロックといわれる。但し、実際には有効なブロックの先頭から2つのブロックがブートブロックとして規定されるようになっており、必ずしもブートブロックがブロック0、1である保証はない。そして、残りのブロックが、ユーザデータが格納されるユーザブロックとなる。

【0022】1ブロックは、図3（d）に示すようにして、ページ0～mにより分割される。1ページの容量は、図3（e）に示すように、512バイトのデータエリアと16バイトの冗長部よりなる、528（=512

+16) バイトの固定長とされる。なお、冗長部の構造については図3 (f) により後述する。また、1ブロック内のページ数としては、1ブロックの容量が8KBの場合には16ページ、16KBの場合には32ページとなる。

【0023】このような、図3 (d) (e) に示されるブロック内のページ構造は、上記ブートブロックとユーザブロックとで共通である。また、フラッシュメモリでは、データの読み出し、及び書き込みはページ単位で行われ、データの消去はブロック単位で行われるものとされる。そして、データの書き込みは、消去済みのページに対してしか行われぬものとされている。従って、実際のデータの書き換えや書き込みは、ブロック単位を対象として行われることになる。

【0024】先頭のブートブロックは、図3 (b) に示すように、ページ0に対してヘッダーが格納され、ページ1には初期不良データの位置(アドレス)を示す情報が格納される。また、ページ2にはCIS/IDSといわれる情報が格納される。2つめのブートブロックは図3 (c) に示すように、ブートブロックとしてのバックアップのための領域とされている。

【0025】図3 (e) に示された冗長部(16バイト)は、図3 (f) に示す構造を有する。この冗長部は、図のように先頭の第0バイト~第2バイトの3バイトが、データエリアのデータ内容の更新に応じて書き換えが可能なオーバーライトエリアとされる。このオーバーライトエリアのうち、第0バイトにはブロックステータスが格納され、第1バイトにはデータステータスが格納される(Block Flag Data)。また、第2バイトの上位の所定ビットを利用して変換テーブルフラグ(Page Data Status1)が格納される。

【0026】原則として第3バイト~第15バイトは、その内容が現ページのデータ内容に応じて固定とされ、書き換えが不可とされる情報が格納される領域となる。第3バイトには管理フラグ(Block Info)が格納され、第4、第5バイトから成る2バイトの領域には、後述する論理アドレス(Logic Address)が格納される。第6~第10バイトの5バイトの領域は、フォーマットリザーブの領域とされ、続く第11、第12バイトの2バイトが、上記フォーマットリザーブに対して誤り訂正を施すための分散情報ECCを格納する領域とされる。残る第13~第15バイトには、図3 (e) に示すデータエリアのデータに対して誤り訂正を行うためのデータECCが格納される。

【0027】上記図3 (f) に示した冗長部の第3バイトに格納される管理フラグは、図4に示すようにして、ビット7~ビット0の各ビットに、その内容が定義されている。ビット7、6、及びビット1、0はリザーブ(未定義)領域とされている。ビット5は現ブロックに対してのアクセス許可の「有効」('1'; Free) /

「無効」('0'; Read Protected)を示すフラグが格納される。ビット4には現ブロックについてのコピー禁止指定('1'; OK / '0'; NG)についてのフラグが格納される。

【0028】ビット3は変換テーブルフラグとされる。この変換テーブルフラグは、現ブロックが後述する論理-物理アドレス変換テーブルであるのか否かを示す識別子であり、このビット3の値が'0'とされていれば、現ブロックは論理-物理アドレス変換テーブルであることが識別され、'0'であれば無効となる。つまり、現ブロックは論理-物理アドレス変換テーブルではないことが識別される。

【0029】ビット2はシステムフラグが格納され、'1'であれば現ブロックがユーザブロックであることが示され、'0'であればブートブロックであることが示される。

【0030】ここで、セグメント及びブロックと、フラッシュメモリ容量との関係を図9(左3列を参照)により説明しておく。板状メモリ1のフラッシュメモリ容量としては、4MB、8MB、16MB、32MB、64MB、128MBの何れかであるものとして規定されている。そして、最も容量の小さい4MBの場合であると、1ブロックは8KBと規定された上で、そのブロック数としては512個とされる。つまり、4MBはちょうど1セグメントの容量を有するものとされる。そして、4MBの容量であれば、同様に1ブロック=8KBの容量が規定された上で、2セグメント=1024ブロックとなる。なお、前述したように、1ブロック=8KBであれば、1ブロック内のページ数は16ページとなる。但し16MBの容量では、1ブロックあたりの容量として8KBと16KBの両者が存在することが許可されている。このため、2048ブロック=4セグメント(1ブロック=8KB)のものと、1024ブロック=2セグメント(1ブロック=16KB)のものと2種類が在ることになる。1ブロック=16KBの場合には、1ブロック内のページ数は32ページとなる。

【0031】また、32MB、64MB、128MBの容量では、1ブロックあたりの容量は16KBのみであるとして規定される。従って、32MBでは2048ブロック=4セグメントとなり、64MBでは4096ブロック=8セグメントとなり、128MBでは8192ブロック=16セグメントとなる。

【0032】2-3. 物理アドレス及び論理アドレスの概念

次に、上述したようなフラッシュメモリの物理的データ構造を踏まえたうえで、図5に示すデータ書き換え動作に従って、本例のファイルシステムにおける物理アドレスと論理アドレスの概念について説明する。

【0033】図5 (a) には、或るセグメント内から4つのブロックを抜き出して、これを模式的に示してい

る。各ブロックに対しては物理アドレスが付される。この物理アドレスはメモリにおけるブロックの物理的な配列順に従って決まるもので、或るブロックとこれに対応付けされた物理アドレスとの関係は不変となる。ここでは、図5(a)に示す4ブロックに対して、上から順に物理アドレスの値として、105, 106, 107, 108が付されている。なお、実際の物理アドレスは2バイトにより表現される。

【0034】ここで、図5(a)に示すように、物理アドレス105, 106で示されるブロックがデータの記憶されている使用ブロックで、物理アドレス107, 108で示されるブロックがデータが消去、即ち、未記録領域とされた未使用ブロックとなっている状態であるとする。

【0035】そして、論理アドレスであるが、この論理アドレスは、ブロックに対して書き込まれたデータに付随するようにして割り振られるアドレスとされる。そして、この論理アドレスが、後述するFATファイルシステムが利用するアドレスとされている。図5(a)では、4つの各ブロックに対して、上から順に論理アドレスの値として、102, 103, 104, 105が付されている状態が示されている。なお、論理アドレスも実際には2バイトにより表現されるものである。

【0036】ここで、上記図5(a)に示す状態から、例えば物理アドレス105に格納されているデータの更新として、内容の書き換え又は一部消去を行うとする。このような場合、フラッシュメモリのファイルシステムでは、同じブロックに対して更新したデータを再度書き込むことはせずに、未使用のブロックに対してその更新したデータを書き込むようにされる。つまり、例えば図5(b)に示すようにして、物理アドレス105のデータは消去したうえで、更新されたデータはこれまで未使用ブロックであった物理アドレス107で示されるブロックに書き込むようにされる(処理^②)。

【0037】そして、処理^②として示すように、データ更新前(図5(a))の状態では物理アドレス105に対応していた論理アドレス102が、更新されたデータが書き込まれたブロックの物理アドレス107に対応するように、論理アドレスについての変更を行うものである。これに伴って、データ更新前は物理アドレス107に対応していた論理アドレス104については、物理アドレス105に対応するように変更されている。

【0038】つまり、物理アドレスはブロックに対して固有に付されるアドレスであり、論理アドレスは、一旦ブロックに対して書き込まれたデータに付随するようにしてついて回る、ブロック単位の書き込みデータに固有となるアドレスであるとすることができる。

【0039】このようなブロックのスワップ処理が行われることで、或る同一の記憶領域(ブロック)に対して繰り返し集中的にアクセスされることが無くなり、書き

換え回数の上限があるフラッシュメモリの寿命を延ばすことが可能となる。そして、この際に論理アドレスを上記処理^②のようにして扱うことで、ブロックのスワップ処理によって更新前と更新後のデータとで書き込まれるブロックの移動があるようにされても、FATからは同一のアドレスが見えることになり、以降のアクセスを適正に実行することができるものである。なお、後述する論理-物理アドレス変換テーブル上での更新のための管理を簡略にすることなどを目的として、ブロックのスワップ処理は、1セグメント内で完結するものとして規定されている。逆に言えば、ブロックのスワップ処理はセグメント間で跨るようにしては行われない。

【0040】2-4. 論理-物理アドレス変換テーブル
上記図5による説明から分かるように、ブロックのスワップ処理が行われることで、物理アドレスと論理アドレスの対応は変化する。従って、フラッシュメモリに対するデータの書き込み及び読み出しのためのアクセスを実現するには、物理アドレスと論理アドレスとの対応が示される論理-物理アドレス変換テーブルが必要となる。つまり、論理-物理アドレス変換テーブルをFATが参照することで、FATが指定した論理アドレスに対応する物理アドレスが特定され、この特定された物理アドレスにより示されるブロックにアクセスすることが可能になるものである。逆に言えば、論理-物理アドレス変換テーブルが無ければ、FATによるフラッシュメモリへのアクセスが不可能となる。

【0041】従来では、例えばセット本体に対して板状メモリ1が装着されたときに、セット本体側のマイクロプロセッサが板状メモリ1の記憶内容をチェックすることで、セット本体側で論理-物理アドレス変換テーブルの構築を行い、更にこの構築された論理-物理アドレス変換テーブルをセット本体側のRAMに格納するようにしていた。つまり、板状メモリ1内には、論理-物理アドレス変換テーブルの情報は格納されてはいなかった。これに対して本例では、以降説明するように板状メモリ1に対して、論理-物理アドレス変換テーブルを格納するように構成している。

【0042】図6は、本例の板状メモリ1に対して格納される論理-物理アドレス変換テーブルの構築形態を概念的に示すものである。つまり、本例では、例えば論理アドレスの昇順に従って、これに対応する2バイトの物理アドレスを格納するようにしたテーブル情報を論理-物理アドレス変換テーブルとして構築するようにされる。なお、前述したように、物理アドレス、及び論理アドレスは共に2バイトで表現される。これは、128MBの最大容量のフラッシュメモリの場合には8192個のブロックが存在するため、最大で、この8192個のブロック数をカバーできるだけのビット数が必要とされることに基づく。このため、図6において例示している物理アドレスと論理アドレスとについても、実際に即し

て2バイトで表現している。但し、ここでは、この2バイトを16進数により表記している。つまり、「0x」によりその後に続く値が16進法表記であることが示される。なお、この「0x」により16進数であることを表す表記は、以降の説明において16進数を表記する場合にも同様に用いることとする。(但し、表記の煩雑化を防ぐため「0x」を省略している図面もある。)

【0043】図7に、上記図6に示した概念に基づく論理-物理アドレス変換テーブルの構造例を示す。論理-物理アドレス変換テーブルは、フラッシュメモリの最後のセグメント内の或るブロックに対して、図7に示すようにして格納される。先ず図7(a)に示すように、ブロックを分割するページのうち、ページ0, 1からなる2ページの領域がセグメント0用の論理-物理アドレス変換テーブルとして割り当てられる。例えば、図9にて説明したように、フラッシュメモリが4MBの容量であれば1セグメントしか有さないために、このページ0, 1のみの領域が論理-物理アドレス変換テーブルの領域となる。また、例えばフラッシュメモリが8MBの容量であれば2セグメントを有するため、セグメント0用の論理-物理アドレス変換テーブルとして割り当てられるページ0, 1に加え、これに続くページ2, 3の2ページがセグメント1用の論理-物理アドレス変換テーブルとして割り当てられることになる。

【0044】以降、フラッシュメモリの容量の増加に応じて、続く2ページごとにセグメントごとの論理-物理アドレス変換テーブルの割り当て領域が設定されていくことになる。そして、最大の128MBの容量を有する場合であれば16セグメントが存在するため、最大では、セグメント15用までのページが論理-物理アドレス変換テーブルの領域として割り当てられることになる。従って、最大の128MBの容量のフラッシュメモリでは、32ページが使用されることになり、図7(a)に示すページNとしては、最大でN=31となる。これまでの説明から分かるように、論理-物理アドレス変換テーブルは、セグメントごとに管理されるものである。

【0045】図7(b)は、1セグメントあたりの論理-物理アドレス変換テーブルの構造を示すものとして、2ページ分のデータエリアを抜き出して示している。つまり、1ページのデータエリアは512バイト(図3(e)参照)であることから、図7(b)には、1024(=512×2)バイトが展開されている状態が示されている。

【0046】図7(b)に示すように、この2ページ分のデータエリアである1024バイトについて2バイトごとに区切り、この2バイトごとの領域を、先頭から順次、論理アドレス0用、論理アドレス1用・・・、のようにして割付を行い、最後は先頭から991バイト目と992バイト目の2バイトの領域を論理アドレス49

5用の領域として割り付けるように規定を行う。これら2バイトごとの領域に対して、各論理アドレスが対応する物理アドレスを書き込むようにする。従って、本例の論理-物理アドレス変換テーブルでは、実際のデータ更新によるブロックのスワップ処理などにより物理アドレスと論理アドレスの対応が変更された場合には、論理アドレスを基準として、物理アドレスの格納状態が更新されるようにしてテーブル情報の書き換えが行われることになる。

【0047】また、残る993バイト目から最後の1024バイト目までの計32バイトの領域は、余剰ブロックの物理アドレスが格納される領域として割り当てられる。つまり、16個の余剰ブロックの物理アドレスを管理することができる。ここでいう余剰ブロックとは、例えばブロック単位でデータの更新を行う際に書き換え対象となるデータを一時待避させる領域として設定されたいわゆるワークブロックなどを言うものである。

【0048】ところで、1セグメントは512ブロックに分割されているものであると先に説明したのにも関わらず、図7に示したテーブル構造では、管理可能なブロック数が論理アドレス0用～論理アドレス495用の496ブロックとしている。これは、實際上、上記した余剰アドレスが設定されることと、前述したように、フラッシュメモリでは、或ブロック数のディフェクト(使用不可領域)が許可されている。そのため現実には、相当数のディフェクトブロックが存在することに依る。従って、実際には、書き込み又は消去が有効なブロックを管理するのに、496ブロックを管理できるように構成しておけば充分とされるものである。

【0049】そして、このようにして論理-物理アドレス変換テーブルが格納されるブロックについては、これを形成する各ページの冗長部における管理フラグ(図4参照)のデータ内容として、この管理フラグのビット3に対して‘0’がセットされることになる。これにより、当該ブロックが論理-物理アドレス変換テーブルが格納されているブロックであることが示されることになる。

【0050】論理-物理アドレス変換テーブルが格納されるブロックも、論理-物理アドレス変換テーブルの内容の書き換えがあった場合には、例外なく、先に図5にて説明したスワップ処理が行われる。従って、論理-物理アドレス変換テーブルが記録されているブロックは不定であり、或る特定のブロックに論理-物理アドレス変換テーブルを格納するように規定することは出来ない。そこで、FATは、フラッシュメモリにアクセスして上記した管理フラグのビット3が‘0’とされているブロックを検索することで、論理-物理アドレス変換テーブルが格納されているブロックを識別するようにされる。但し、論理-物理アドレス変換テーブルが格納されているブロックの検索がFATによって容易に行われるよう

にすることを考慮して、論理－物理アドレス変換テーブルが格納されているブロックは、そのフラッシュメモリ内における最後のナンバが付されたセグメントに在るように、本例では規定するものとされる。これにより、FATは最後のナンバが付されたセグメントのブロックのサーチだけで、論理－物理アドレス変換テーブルを検索することができる。つまり、論理－物理アドレス変換テーブルを検索するのに、フラッシュメモリの全てのセグメントを検索する必要は無いようにされる。上記図7に示した論理－物理アドレス変換テーブルは、例えば板状メモリ1の製造時において格納するようにされる。

【0051】ここで、再度図9を参照して、フラッシュメモリ容量と論理－物理アドレス変換テーブルのサイズとの関係を説明しておく。上記図7にて説明したように、1セグメントを管理するための論理－物理アドレス変換テーブルのサイズは2ページ分の1024バイト、つまり1KBとなる。従って、図9の最右列に記されているように、フラッシュメモリが4MB（1セグメント）の容量では論理－物理アドレス変換テーブルは1KBのサイズとなる。また、フラッシュメモリの容量が8MB（2セグメント）では論理－物理アドレス変換テーブルは2KB（4ページ）となる。また、フラッシュメモリの容量が16MBの場合、2048ブロック＝4セグメントのものでは論理－物理アドレス変換テーブルは4KB（8ページ）、1024ブロック＝2セグメントのものでは論理－物理アドレス変換テーブルは2KB（4ページ）となる。そして、フラッシュメモリの容量が32MB（4セグメント）では論理－物理アドレス変換テーブルは4KB（8ページ）、フラッシュメモリの容量が64MB（8セグメント）では論理－物理アドレス変換テーブルは8KB（16ページ）となり、フラッシュメモリの容量が最大の128MB（16セグメント）では論理－物理アドレス変換テーブルは16KB（32ページ）となる。

【0052】ところで、例えば従前におけるフラッシュメモリのファイルシステムの構成では、論理－物理アドレス変換テーブル上において、未使用とされる論理アドレスに対応する物理アドレスの領域には、実質的なアドレス値は不確定として格納しないようにされている。一例を挙げると、図8（b）に示す論理－物理アドレス変換テーブルにおいて、例えば論理アドレス0x0000、0x0001、0x0002、0x0003は既に使用されているものとする、これらに対応する物理アドレスの格納領域には、例えばそれぞれ、0x0002、0x0006、0x0007、0x0008という実際に論理アドレス（0x0000～0x0003）に対応してデータが書き込まれた物理アドレスの値が格納されていることになる。これに対して、論理アドレス0x0004は未使用であるとするならば、この論理アドレス0x0004が対応する物理アドレスの格納領域に

は、0xFFFFという無効値（即ち、未使用であることが識別される値）が設定される。

【0053】ここで、上記図8（b）に示す論理－物理アドレス変換テーブルを利用して、論理アドレス0x0004に対応するデータを新規に記録しようとした場合には、例えばFATにより、論理－物理アドレス変換テーブルとは異なる階層で、物理的に未使用とされているブロックの検索を行い、この検索されたブロックに対してデータの書き込みを実行するようにされる。そして、論理－物理アドレス変換テーブルに対しては、そのデータ書き込みが新規に行われたブロックを示す物理アドレスを論理アドレス0x0004に対応する格納領域に対して書き込むようにして、その内容を更新することになる。

【0054】このような論理－物理アドレス変換テーブルの管理形態であると、次のような不都合が生じることが想定される。ここで、セット本体側において扱うべきデータが、動画データや楽曲等の音声データなど、いわゆるリアルタイム性を有する時系列データ（本例でいうストリームデータ）であるとする。そして、セット本体では、入力されてきたストリームデータについてリアルタイムに信号処理を行って、このデータを記録データとして板状メモリ1に記録するような構成が採られているものとする。

【0055】このとき上記図8（b）により説明したような論理－物理アドレス変換テーブルの管理形態を採用しているとすると、データ記録時には、先に述べたような未使用ブロックのサーチが必然となる。しかし、上記したようなストリームデータを記録していく場合には、入力された記録データがオーバーフローしない程度の平均速度で書き込む必要があるため、この際の未使用ブロックのサーチは非常に重い処理となってしまう。つまり、リアルタイム性を有するストリームデータを板状メモリ1に記録することは非常に困難であり、実用上は、リアルタイム性が要求されない文書ファイルや静止画ファイルなどの記録にとどめられてしまう。

【0056】そこで、本例の論理－物理アドレス変換テーブルとしては、未使用の論理アドレスに対応する格納領域に対して、未使用のエリアとして管理されているブロックの物理アドレスを予め割り当てておくようにする。この具体例を図8（a）に示す。図8（a）に示す論理－物理アドレス変換テーブルにおいては、論理アドレス0x0000、0x0001、0x0002、0x0003は使用されているものとされ、これに対応する物理アドレスの格納領域に対して、例えば0x0002、0x0006、0x0007、0x0008という、実際にデータが記録されている物理アドレスの値が格納されている点では図8（b）と同様である。また、論理アドレス0x0004は未使用であるとしている点も図8（b）と同様である。

【0057】但しこの本例の場合、この未使用の論理アドレス0x0004が対応する物理アドレスの格納領域に対しては、0xFFFFという無効値を設定するのではなく、未使用のエリアのブロックを指定する物理アドレスとして、例えば0x0009を格納する。ここでは、未使用のエリアに関して、論理アドレス0x0004に対応する格納領域しか示していないが、他の未使用とされる論理アドレスに対応する格納領域の各々に対しても、同様にして、それぞれ異なる未使用のブロックの物理アドレスが格納されるものである。

【0058】このようにして論理-物理アドレス変換テーブルを構築した場合、論理-物理アドレス変換テーブルに対して既に空きエリアを示す物理アドレスが、論理アドレスと対応付けられて示されているものと見ることができる。これにより、FATでは、論理-物理アドレス変換テーブルを参照することで事前に未使用のブロックの物理アドレスと論理アドレスとの対応を探ることができ、データ記録時においては、図8(b)に示した形態の場合のように、未使用のブロックをサーチする処理を実行する必要がなくなる。つまり、FAT上で空きエリアとして管理される論理アドレスに対応する物理アドレスを論理-物理アドレス変換テーブルを参照することにより得て、この物理アドレスにより示されるブロックに対してアクセスしてデータ書き込みを実行すればよいことになる。このため、セット本体のマイクロコンピュータの処理負担は大幅に軽減され、例えば先に述べたストリームデータの記録も容易に実現可能となるものである。なお、これまでのように文書ファイルや静止画データなど、リアルタイムな処理が要求されないデータ書き込みの際しても、図8(a)に示した本例のファイルシステムとすれば、当然のこととして、これまでよりもデータ書き込みに要する時間は短縮されるものである。

【0059】以上のようにあらかじめ未使用の論理アドレスと物理アドレスについてまでも対応をとったアドレス変換テーブルを、板状メモリ1に記録又は必要に応じて更新することは、次に説明するドライブ装置によって行われることになるが、少なくとも後述する高速書込処理が実行される時点よりも前において、未使用の論理アドレスと物理アドレスが対応づけられた状態とされていればよい。また、実際には、板状メモリ1の使用前の最初のフォーマット処理として、ドライブ装置が、全論理アドレスと全物理アドレスを対応づけたアドレス変換テーブルを作成して、板状メモリ1に記録するとともに、その後、記録/編集動作に応じて、図5で説明したような更新を行うようにすることが考えられる。

【0060】3. ドライブ装置の構成

続いて図10で本例のドライブ装置の構成を説明する。図10は、これまで説明した板状メモリ1に対応してデータの読出・書込・編集を行うことの出来るドライブ装置のセット本体(本体装置)の構成を示している。この

図に示すセット本体100と板状メモリ1とにより、ファイル記憶システムが構成される。なお、セット本体100が、板状メモリ1に対する書込や読出の対象として扱うことのできる主データの種類の多様であり、例えば動画データ、静止画データ、マイクロホン入力音声等のボイスデータ、高音質(HiFi)オーディオデータ(音楽用データともいう)、制御用データなどがある。本例では、ストリームデータの効率的な記録を実現することを目的とするものであるため、主データとしてのストリームデータの1つであるボイスデータを記録/再生するシステムとして説明していくが、セット本体100内に、動画、静止画、音楽等のデータの入出力系/処理系やコンピュータインターフェースを備えることにより、他のストリームデータや、リアルタイム性を要しないデータとしてのデータファイルの記録システムとできることはいうまでもない。

【0061】セット本体100の構成としては、当該セット本体100が着脱可能に装填される着脱機構120が備えられ、この着脱機構120に装填された板状メモリ1と、マイクロコンピュータ109とのデータの授受は、ホストインターフェイスIC101を介することで行われる。

【0062】また、このセット本体100には、例えば、マイクロフォン103が備えられて、このマイクロフォン103により収音された音声は、マイクアンプ104を介して音声信号としてDSP(Digital Signal Processor)102に対して供給される。DSP102では、この入力された音声信号をデジタルオーディオデータに変換して、エンコード処理等をはじめとする所要の信号処理を施し、記録データとしてボイスデータを制御用マイクロコンピュータ109に供給する。マイクロコンピュータ109は、この記録データをホストインターフェイスIC101を介して板状メモリ1に記録するための処理を実行することが可能とされる。

【0063】また、マイクロコンピュータ109は、板状メモリ1に記録されているオーディオデータ又はボイスデータファイルをホストインターフェイスIC101を介して読み出し、この読み出したデータをDSP102に供給する。DSP102では、供給されたデータについて復調処理をはじめとする所要の信号処理を施して、最終的にはアナログ音声信号としてスピーカアンプ105に出力する。スピーカアンプ105では、入力された音声信号を増幅してスピーカ106に出力する。これにより、再生音声出力される。

【0064】また、マイクロコンピュータ109は、表示ドライバ107を制御することで、表示部108に対して、所要の画像を表示させることが可能とされる。例えばユーザーの操作のためのメニューやガイド表示、或いは板状メモリ1に記憶されたファイル内容などの表示が実行される。また、例えば板状メモリ1に対して動画

若しくは静止画の画像データが記録されているとすれば、この画像データを読み出して、表示部108に表示させるようにすることも可能とされる。

【0065】操作部112には、セット本体100に対する各種操作をユーザが行うための各種キーが設けられている。マイクロコンピュータ109は、この操作部112に対して行われた操作に応じたコマンドを受信し、コマンドに応じた所要の制御処理を実行する。操作内容としては、ファイル記録指示、ファイル選択指示、ファイル再生指示、編集指示などが可能とされる。

【0066】この図10に示す構成のセット本体100により、上述したような板状メモリ1に対する記録再生（データ書き込み又は読み出し）を実現するには、ファイルシステム上でFATが参照する論理－物理アドレス変換テーブルが必要となるのは前述したとおりである。そこで図11に、上記図10に示す構成に基づくセット本体100のマイクロコンピュータ109と板状メモリ1に格納されている論理－物理アドレス変換テーブル間とのインターフェイスを概念的に示す。例えば、本例の板状メモリ1が装填されたときには、マイクロコンピュータ109は、ホストインターフェイスIC101を介して、板状メモリ1に格納されている論理－物理アドレス変換テーブルのうちから必要なデータ部分を読み込んで、内部のRAM111に格納すればよいことになる。

【0067】なおこのことから、本例では、セット本体100のマイクロコンピュータ101の処理として、論理－物理アドレス変換テーブルを構築する処理は必要なく、このため、マイクロコンピュータ109による論理－物理アドレス変換テーブル構築処理のための待ち時間はなくなり、例えば板状メモリ1を装着したときに行われるファイルシステムの立ち上げ処理は、従来よりもはるかに短時間で済むことになる。

【0068】更に、図8(a)にて説明したように、未使用のブロックの物理アドレスを未使用の論理アドレスと対応させた論理－物理アドレス変換テーブルとすることで、FATによる未使用のブロックへのアクセスは、従来よりも軽い処理で迅速に実行されることになる。特に、図10に示したように、音声データ等のリアルタイムな処理が要求されるデータを記録する構成をセット本体100が採るような場合には有効となるものである。

【0069】なお、図10に示したセット本体100の構成はあくまでも一例であり、これに限定されるものではない。つまり、板状メモリ1に対応してデータの書込が可能な構成を採る限りは、どのようなタイプの記録装置とされていても構わないものである。

【0070】4. FAT構造

図2のファイルシステム階層で説明したように、ファイル管理処理はFATにより行われることになる。即ち図

10に示した構成のドライブ装置により、板状メモリ1に対する記録再生を実現するには、アプリケーション処理での要求に伴ってFATによるファイル記憶位置管理が参照され、さらに上述した論理－物理アドレス変換が行われて実際のアクセスが行われることになる。ここで、FATの構造について説明しておく。

【0071】図12はFATによる管理構造の概要を示している。なお、本例ではFAT及び論理－物理アドレス変換テーブルは板状メモリ1内に格納されることになるが、図12に示すFAT構造が、板状メモリ1内での管理構造となるものである。

【0072】そして、図示するようにFAT管理構造は、パーティションテーブル、空き領域、ブートセクタ、FAT、FATのコピー、ルートディレクトリ、データ領域から成る。データ領域には、クラスタ2、クラスタ3・・・として単位データを示しているが、このクラスタとは、FATで扱う管理単位としての1データ単位となる。一般にFATでは、クラスタサイズは標準で4Kバイトとされるが、このクラスタサイズは512バイト～32Kバイトの間で2のべき乗の大きさをとることができる。本例の板状メモリ1では、上述したように1つのブロックが8Kバイト又は16Kバイトとされるが、1ブロック＝8Kバイトとされる板状メモリ1の場合は、FATで扱うクラスタは8Kバイトとされる。また1ブロック＝16Kバイトとされる板状メモリ1の場合は、FATで扱うクラスタは16Kバイトとされる。即ち、8Kバイト又は16KバイトがFAT管理上でのデータ単位であり、かつ板状メモリ1でのブロックとしての1つのデータ単位とされる。なお、従って板状メモリからみれば、FATで扱われるクラスタサイズ＝その板状メモリのブロックサイズとなる。このため、本例の以降の説明については、簡略化のためにブロック＝クラスタとして考えることとする。

【0073】そして図12左側にブロックナンバとして $x \cdots (x+m-1)$ 、 $(x+m)$ $(x+m+1)$ $(x+m+2) \cdots$ と示したが、例えばこのように各ブロックにおいてFAT構造を構築する各種データは記憶されることになる。なお、実際には必ずしもこのように物理的に連続する各ブロックに各情報が記憶されるものではない。

【0074】FAT構造において、まずパーティションテーブルには、FATパーティション（最大2Gバイト）の先頭と終端のアドレスが記述されている。ブート領域には、いわゆる12bitFAT、16bitFATの別や、FAT構造（大きさ、クラスタサイズ、各領域のサイズなど）が記述される。

【0075】FATは、後述するように各ファイルを構成するクラスタのリンク構造を示すテーブルとなり、またFATについては続く領域にコピーが記述される。ルートディレクトリには、ファイル名、先頭クラスタ番

号、各種属性が記述される。これらの記述は1つのファイルにつき32バイト使用される。

【0076】FATにおいては、FATのエントリとクラスタは1対1で対応しており、各クラスタのエントリにはリンク先、つまり後に続くクラスタの番号が記述される。つまり、複数のクラスタ(=ブロック)で形成されている或るファイルについてみると、まずディレクトリによって先頭のクラスタ番号が示され、FATにおけるその先頭クラスタのエントリには、次のクラスタ番号が示される。さらに次のクラスタ番号のエントリには、さらに次のクラスタ番号が示される。このようにクラスタのリンクがFATに記述される。

【0077】図13はこのようなリンクの概念を模式的に示している(数値は16進値)。例えば2つのファイル「MAIN. C」「FUNC. C」が存在するとすると、ディレクトリにはこの2つのファイルの先頭クラスタ番号が例えば「002」「004」と記述される。そしてファイル「MAIN. C」については、クラスタ番号「002」のエントリに次のクラスタ番号「003」が記述され、またクラスタ番号「003」のエントリに次のクラスタ番号「006」が記述される。さらに、クラスタ番号006がこのファイル「MAIN. C」の最後のクラスタであるとする、クラスタ番号「006」のエントリには、最後のクラスタであることを示す「FFF」が記述される。これによりファイル「MAIN. C」がクラスタ「002」→「003」→「006」という順番で記憶されている。即ち、仮にクラスタ番号と板状メモリ1でのブロック番号が一致していると仮定すると、ファイル「MAIN. C」は、板状メモリ1内でブロック「002」「003」「006」に記憶されていることが表現されている。(但し、FATで扱うクラスタは、上述のように論理アドレスで扱うものとなるため、ブロックの物理アドレスとそのまま一致するものではない)

【0078】また同様にファイル「FUNC. C」については、FATにより、クラスタ「004」→「005」に記憶されていることが表現される。

【0079】なお、未使用のブロックに対応するクラスタについては、そのエントリは「000」とされる。

【0080】ところでルートディレクトリの領域に記憶される各ファイルのディレクトリにおいては、図13に示した先頭クラスタ番号だけでなく、例えば図14のように各種データが記述される。即ちファイル名、拡張子、属性、変更時刻情報、変更日付情報、先頭クラスタ番号、ファイルサイズが、それぞれ図示するバイト数で記述される。

【0081】また或るディレクトリの下層となるサブディレクトリについては、図12のルートディレクトリの領域ではなく、データ領域に記憶される。つまりサブディレクトリは、ディレクトリ構造を持つファイルとして

扱われる。そしてサブディレクトリの場合はサイズは無制限とされ、また自分自身へのエントリと親ディレクトリへのエントリが必要になる。

【0082】図15に、或るルートディレクトリ内にファイル「DIR1」(属性=ディレクトリ:つまりサブディレクトリ)があり、さらにその中にファイル「DIR2」(属性=ディレクトリ:つまりサブディレクトリ)があり、さらにその中にファイル「FILE」が存在する場合の構造例を示している。つまりルートディレクトリの領域には、サブディレクトリであるファイル「DIR1」としての先頭クラスタ番号が示され、上述したFATにより、クラスタX、Y、Zがリンクされている状態となる。この図からわかるように、サブディレクトリ「DIR1」「DIR2」についてはファイルとして扱われてFATのリンクに組み込まれる。

【0083】以上の説明を図22を用いてまとめる。図22にFAT管理による管理方法を説明する。図22は、板状メモリ1内の模式図(メモリマップ)を示しており、上からパーティションテーブル、空き領域、ブートセクター、FAT領域、FATのコピー領域、ルートディレクトリ領域、サブディレクトリ領域、データ領域が積層されている。なお、このメモリマップは上述した論理-物理アドレス変換テーブルに基づく、論理アドレス→物理アドレス変換後のメモリマップである。

【0084】ブートセクター、FAT領域、FATのコピー領域、ルートディレクトリ領域、サブディレクトリ領域、データ領域を全部まとめてFATパーティション領域と称する。

【0085】上記パーティションテーブルにはFATパーティション領域のはじめと終わりのアドレスが記録されている。なお通常フロッピーディスクで使用されているFATには上記パーティションテーブルは備えられていない。またメモリ内の先頭領域にはパーティションテーブル以外のものは置かないために図示するように空きエリアが生ずる。

【0086】ブートセクターには12ビットFAT/16ビットFATであるかの別に応じてFAT構造の大きさ、クラスタサイズ、それぞれの領域のサイズが記録されている。FATはデータ領域に記録されているファイル位置を管理するものである。FATのコピー領域はFATのバックアップ用の領域である。

【0087】ルートディレクトリはファイル名、先頭クラスタアドレス、各種属性が記録されており、1ファイルにつき32バイト使用する。サブディレクトリはディレクトリというファイルの属性のファイルとして存在しており、この図22の例ではPBLIST. MSV, CAT. MSV, DOG. MSV, MAN. MSVという4つのファイルが存在する。

【0088】このサブディレクトリにはファイル名とFAT上の記録位置が管理されている。即ち、図22にお

いては、CAT、MSVというファイル名が記録されているスロットには”5”というFAT上のアドレスが管理されており、DOG、MSVというファイル名が記録されているスロットには”10”というFAT上のアドレスが管理されている。更に、MAN、MSVというファイル名が記録されているスロットには”110”というFAT上のアドレスが管理されている。

【0089】クラスタ2以降が実際のデータ領域で、このエリアに本例ではADPCMで圧縮処理されたボイスデータが記録される。本例では、クラスタ5～クラスタ8にCAT、MSVというファイル名のADPCMで圧縮処理されたボイスデータが記録され、クラスタ10～クラスタ12にDOG、MSVというファイル名の前半パートであるDOG-1がADPCMで圧縮処理されたボイスデータが記録され、クラスタ100～クラスタ101にDOG、MSVというファイル名の後半パートであるDOG-2がADPCMで圧縮処理されたボイスデータが記録されている。更に、クラスタ110～クラスタ111にMAN、MSVというファイル名のADPCMで圧縮処理されたボイスデータが記録されている。またデータ領域上のEmptyとかかれた領域は記録可能領域である。

【0090】この例において、DOG、MSVというファイル名のファイルは、単一のファイルが2分割されて離散的に記録されている例となっている。

【0091】クラスタ200以降はファイルネームを管理する領域であり、クラスタ200にはCAT、MSVというファイルが、クラスタ201にはDOG、MSVというファイルが、クラスタ202にはMAN、MSVというファイルが記録されている。ファイル順を並び替える場合には上記クラスタ200以降で並び替えを行えばよい。

【0092】最初、このような板状メモリ1が挿入された場合には、ドライブ装置は、先頭のパーティションテーブルを参照してFATパーティション領域のはじめと終わりのアドレスを把握する。そしてブートセクタの再生を行った後にルートディレクトリ、サブディレクトリの再生を行う。そしてサブディレクトリに記録されている再生管理情報である、PBLIST、MSFが記録されているスロットを検索して、PBLIST、MSFが記録されているスロットの終端部のアドレスを参照する。図22の例では、PBLIST、MSFが記録されているスロットの終端部には”200”というアドレスが記録されているので、これに基づいてクラスタ200を参照することになる。

【0093】上述したようにクラスタ200以降はファイル名を管理するとともにファイルの再生順を管理する領域であり、この例の場合にはCAT、MSVというファイルが1番目となり、DOG、MSVというファイルが2番目となり、MAN、MSVというファイルが3番

目となる。

【0094】ここでクラスタ200以降を全て参照したら、サブディレクトリに移行して、上記CAT、MSV及びDOG、MSV及びMAN、MSVという名前のファイル名と合致するスロットを参照する。図22においては、CAT、MSVというファイル名が記録されたスロットの終端には”5”というアドレスが記録され、DOG、MSVというファイルが記録されたスロットの終端には”10”というアドレスが記録され、MAN、MSVというファイルが記録されたスロットの終端には110というアドレスが記録されている。

【0095】ここで、CAT、MSVのスロットにおける”5”というアドレスに基づいて、FAT上のエントリーアドレスを検索すると”6”というクラスタアドレスがエントリーをされており、”6”というエントリーアドレスを参照すると”7”というクラスタアドレスがエントリーされており、”7”というエントリーアドレスを参照すると”8”というクラスタアドレスがエントリーされており、”8”というエントリーアドレスを参照すると”FFF”という終端を意味するコードが記録されている。よって、CAT、MSVというファイルは、クラスタ5、6、7、8の領域を使用しており、データ領域のクラスタ5、6、7、8を参照することでCAT、MSVというADPCMデータが実際に記録されている領域をアクセスすることができる。

【0096】また離散的に記録されているDOG、MSVというファイルを検索する場合は以下になる。DOG、MSVというファイルが記録されたスロットの終端には”10”というアドレスが記録されている。ここで、上記”10”というアドレスに基づいて、FAT上のエントリーアドレスを検索すると”11”というクラスタアドレスがエントリーをされており、”11”というエントリーアドレスを参照すると”12”というクラスタアドレスがエントリーされており、”12”というエントリーアドレスを参照すると”100”というクラスタアドレスがエントリーされており、”100”というエントリーアドレスを参照すると”101”というクラスタアドレスがエントリーされており、”101”というエントリーアドレスを参照するとFFFという終端を意味するコードが記録されている。

【0097】よって、DOG、MSVというファイルは、クラスタ10、11、12、100、101というクラスタ領域を使用しており、データ領域のクラスタ10、11、12を参照することでDOG、MSVというファイルの前半パートに対応するADPCMデータが実際に記録されている領域をアクセスすることができる。更に、データ領域のクラスタ100、101を参照することでDOG、MSVというファイルの後半パートに対応するADPCMデータが実際に記録されている領域をアクセスすることができる。

【0098】また、上記MAN、MSVのスロットにおける“110”というアドレスに基づいて、FAT上のエントリーアドレスを検索すると“111”というクラスタアドレスがエントリーをされており、“111”というエントリーアドレスを参照すると“FFF”という終端を意味するコードが記録されている。よって、MAN、MSVというファイルは、クラスタ110、111という領域を使用していることがわかる。

【0099】以上のように板状メモリ1上に記録されたデータファイルをFATを参照して再生することができ、またデータファイルが離散的に記録されていても、それらを連結してシーケンシャルに再生することが可能となる。

【0100】5. ストリームデータの書込処理

続いて本例の最も特徴的な動作となる、ストリームデータの板状メモリ1への記録処理について説明していく。上述したように、板状メモリ1で扱うことのできる主データとしては、動画データ、静止画データ、ボイスデータ、高音質なオーディオデータ（音楽用データ）、制御用データなどがあるが、例えばボイスデータ、動画データ、オーディオデータ等のリアルタイム性を有するストリームデータを記録する場合には、より高いビットレートのストリームデータにも対応できるように、効率的な記録処理がおこなわれることが必要である。このために本例で実行するデータ書込処理を、図32により前述した書込処理と区別して高速書込処理と呼ぶこととし、図16～図20により説明する。

【0101】図16、図17は、ドライブ装置のマイクロコンピュータ109の制御により実行される処理のフローチャートであり、また図18は板状メモリ1上でのストリームデータ（ボイスデータ等）のファイルの記録状態を概念的に示すものである。さらに、図19、図20は記録処理前後でのFAT内容を示している。

【0102】或る板状メモリ1をドライブ装置に装填し、記録操作をおこなうことで、ユーザーはマイクロフォン103により集音される音声を音声データとして板状メモリ1に記憶させていくことができる。

【0103】ユーザーによりこのような記録動作が指示されたら、マイクロコンピュータ109は図16のステップF101として、まず高速処理マーキング作成を行う。これは、高速書込を行うファイルとして、仮にルートディレクトリに特別なファイルエントリを作成する処理となる。そしてそのファイルエントリに応じて、ステップF102で高速ファイルオープンを行なう。これは高速なデータ書き込みに先んじて行われる必要のある処理であり、図17で後に詳細に説明する書き込みが行われるクラスタに対する処理である。ステップF102の高速処理マーキング作成の処理は図31に詳しく示される。なお、作成するファイルエントリについては「temp temp」という暫定的なファイル名と

し、これについては図23～図30を用いた説明において後述する。

【0104】まずステップF130でFATで管理されている未使用クラスタのうちでクラスタ2以降の、その時点でもっとも若いクラスタナンバを変数CL(x)にセットしている。このとき論理物理アドレス変換変換テーブルの内容に基づいて板状メモリ1内のフラッシュメモリのブロック番号に変換しておく。ステップF131でその次に若い未使用クラスタナンバを変数CL(y)にセットしている。このときもステップF130と同様に物理アドレスに変換しておく。次にステップF132においてステップF130で得られたフラッシュメモリの物理アドレスに対応するブロックのデータの消去を行い、ステップF133ではステップF131で得られたフラッシュメモリの物理アドレスに対応するブロックのデータを消去する。なおここまでで得られた物理アドレスは以降に説明するステップで利用されるため保持される。

【0105】以上の図31のように、図16のステップF102の処理を終えたら、続いて図16のステップF103、F104として、入力されるボイスデータの記録が完了するまで（例えばユーザーが記録停止操作を行う直前までのボイスデータの書込が終了するまで）、板状メモリ1に対してクラスタ単位でストリームデータの書込を続ける。

【0106】ステップF103の処理は図17に詳しく示される。すなわちステップF103では、時間的に連続して入力されてくるストリームデータについて、まず図17のステップF120でクラスタ単位のデータ量の取り込みを行い、クラスタ単位のデータ量の取り込みが完了したら、ステップF121でクラスタCL(x)に対応する板状メモリ1のブロックに取り込まれたクラスタ単位データ量のストリームデータの書き込みを行う。このときに使用されるクラスタCL(x)の値は例えば最初のクラスタ単位量のストリームデータの書き込みであった場合は、図16のステップF102、より詳しくは図31のステップF130で得た値である。また、それ以降のストリームデータ書き込みである場合には次に説明するステップF122で得られた値になる。

【0107】次にステップF122においては、新たなクラスタCL(x)として、次にストリームが書き込まれるクラスタの板状メモリ1上での物理ブロックアドレスをCL(y)の値のコピーによって設定する。更に次のステップF123においては、次の次にストリームデータが書き込まれるクラスタであるFATファイル管理された空きクラスタのうちで次に若いクラスタナンバを取得し論理物理アドレス変換テーブルによって板状メモリ1における物理ブロックナンバを変数CL(y)にセットする。そして次のステップF124においてステップF123で得られた板状メモリ1の物理ブロック番号

のブロックのデータを消去している。この処理により次にストリームデータが書き込まれるクラスタの次のクラスタが消去されることになる。

【0108】なお、ここで説明している例は、説明を単純化するためにデータの取り込みやフラッシュメモリへの書き込みの単位を1クラスタすなわちフラッシュメモリ上での1ブロックとしたものである。しかしながら、「2-2. 物理的データ構造」で説明したように、板状メモリ1内のフラッシュメモリへの書き込みの単位はページとなるので、例えばストリームデータの取り込みをフラッシュメモリの1ページに対応する量とし、書き込みも1ページ単位で書き込む用にしても良い。その場合は、1クラスタつまり1ブロック分のデータを書き込むためには、1ブロックが16ページの場合であれば16回書き込みのページを変えながら書き込み処理を繰り返し、1ブロックが32ページの場合であれば32回書き込みのページを変えながら書き込み処理を繰り返すこととなる。そして図16のステップF104による評価を行うことも可能であり、また、1ページの書き込みごとにステップF104に相当する入力されるデータの終了判定を行うようにしても良い。

【0109】以上の図17の処理が、図16のステップF103で実行される。なお、この処理では、空きクラスタ（ブロック）のうちで最も若いクラスタナンバ（論理アドレス）のブロックにストリームデータの書込を行い、次に若いクラスタナンバのブロックについてデータ消去を行うようにしているが、これは後述する復活処理を可能とするためのものであり、処理の意味については後に説明する。

【0110】この図16のステップF103の処理（＝図17の処理）は、ストリームデータの記録が完了するまで、クラスタ単位のデータが取り込まれる毎に繰り返し実行されることになるが、従ってその都度、各時点で最も若いクラスタナンバのブロックに書込が行われ、続いて若いクラスタナンバのブロックに対する消去が行われていくことになる。

【0111】これは、実際の書き込みにおいては先に説明したようにデータの書き込みは1ページ単位での書き込みをしていくわけであるが、1クラスタ分のデータつまり16ページまたは32ページのデータを書き終わる前に次ぎに書き込むべきクラスタが消去されている必要があるからである。フラッシュメモリの消去の単位は先に説明した通りクラスタであるが、1クラスタの消去に要する時間に比べて1ページのデータを書く時間が短いので、最後のページの書き込みが始まるよりも十分に前である必要があることと処理の簡易化をはかるために常に1クラスタ先を消去しておくのである。また、先のステップF102について図31で説明したが、ステップF102では、図17のステップF122、F123、F124と同様の意味の処理とクラスタCL(x)

のデータの消去をしている。この処理をすることによって最初にストリームデータの書き込まれるクラスタCL(x)とその次に書き込まれるクラスタつまり次に若い番号のクラスタCL(y)が消去された状態になるのである。

【0112】上記のステップF103の処理がストリームデータの終了まで継続されるが、全ストリームデータの書込が完了したら、処理はステップF104からF105に進み、ストリームデータの記録動作に応じてFATを更新する。ここでは、FATのエントリを更新して今回の記録に応じたリンク構造が記録されるようにするとともに、ディレクトリエントリの更新、及びステップF101で行った高速処理マーキングの消去を行うことになる。なお後述するが、このときに、例えば「temp temp. tmp」という暫定ファイル名を消去し、記録したストリームデータに本来付与されるファイル名が記録され、ディレクトリ構成内に配される。これによって、ステップF103で書き込んでいったストリームデータがFAT上で適正に管理された有効なファイルデータとなる。FAT更新を終えたら、ステップF106でファイルクローズを行なって処理を終える。

【0113】以上の処理により高速書込処理が完了するわけであるが、この処理によって記録されるストリームデータファイル及びFAT更新状況を図18～図20で述べる。今回の記録前の時点において、図18(a)に示すファイルF1が、板状メモリ1に既に記録されていたとする。そして仮にこのファイルF1は、クラスタCL(2)、CL(3)、CL(4)、CL(6)、CL(8)、CL(9)に記憶されていたものとする。なお実際にはこのクラスタナンバのそれぞれに対応する物理アドレスのブロックに記録されているものであり、これまでの説明からわかるように、実際の記録ブロックの物理アドレスはアドレス変換テーブルに示される。また従って、図18は1つのファイルが物理的に連続したブロックに記録されている状態を表すものではない。

【0114】この場合、このファイルF1のクラスタリンクを示すFATは図19のようになっている。即ち図示していないが図13で説明したディレクトリエントリによってファイルF1の先頭のクラスタがクラスタCL2とされ、図19のFATのクラスタCL(2)のエントリには、「003」と記憶されている。つまりクラスタCL(2)がクラスタCL(3)にリンクすることが示される。またクラスタCL(3)のエントリには「004」と記憶され、クラスタCL(4)にリンクすることが示される。以降同様にリンク状態が記録され、最後のクラスタCL(9)のエントリには最終クラスタであることを示す値「FFF」が記憶される。このようなFATによって、図18(a)のファイルF1がクラスタCL(2)、CL(3)、CL(4)、CL(6)、CL(8)、CL(9)に、その順序で格納されているこ

とが管理される。そしてこの状態では、クラスタCL (5)、CL (7)、及びクラスタCL (A)以降は「000」つまり未使用とされている。

【0115】ここでユーザーが新たに記録を指示し、上記高速書込処理により、新たなストリームデータファイルとして図18 (b) のファイルF 2が記録されたとする。この場合、ストリームデータの記録に7クラスタを用いたとすると、ファイルF 2は、クラスタCL (5)、CL (7)、CL (A)、CL (B)、CL (C)、CL (D)、CL (E) に記録されることになる。

【0116】即ち図17に示した処理により、まず最初のクラスタ単位のデータは、最も若いクラスタナンバの空きクラスタに記録されるため、図19からわかるようにクラスタCL (5) が用いられることになる。またこのとき、次にさらにその次に若いクラスタ (に対応するブロック) がデータ消去されるが、この場合はクラスタCL (7)、CL (A) が消去対象となる。さらに次のクラスタ単位のストリームデータは、同じくその時点で最も若い空きクラスタが用いられることになるため、今度はクラスタCL (7) が用いられる。またこのとき、次にさらにその次に若いクラスタであるクラスタCL (A)、CL (B) が消去対象となる。上述した処理により以上の動作が行われていくことで、ステップF 104で終了と判断された時点では、図18 (b) のようにクラスタCL (5)、CL (7)、CL (A)、CL (B)、CL (C)、CL (D)、CL (E) のそれぞれに対応する物理アドレスのブロックに、ストリームデータが記録された状態となる。但し、この時点ではFATが更新されておらず図19のままであるため、ステップF 105としてFATのリンク及びディレクトリエントリの更新が行われ、ファイルF 2に対応するクラスタリンクが形成される。即ち、ファイルF 2についてのディレクトリエントリが形成され、先頭のクラスタがクラスタCL (5) と指示されるとともに、図20に示すように、FATにおいてクラスタCL (5) のエントリには「007」、クラスタCL (7) のエントリには「00A」、クラスタCL (A) のエントリには「00B」……クラスタCL (E) のエントリには「FFF」と記録される。これによって、高速書込処理により記録されたファイルF 2が、FATによって管理された状態となり、つまり有効なファイルデータとなる。

【0117】これまでの説明について、図23～図30を用いて補足説明する。図23は新しくデータを書き込む前のディレクトリを示している。ルートディレクトリの下には「VOICE」と名付けられたサブディレクトリのほか図示しないサブディレクトリが存在している。また、「VOICE」ディレクトリの下には「FOLDER1」と名付けられたサブディレクトリと図示しないその他のサブディレクトリが存在している。さらに「F

OLDER1」ディレクトリの下には既に記録された「98120100.MSV」と言うファイルが存在している。

【0118】図26は、図23に対応したFATファイルシステム上での各データの配置を示している。ルートディレクトリには「VOICE」という名称のディレクトリが存在することが示されている。そしてサブディレクトリには図示しない「FOLDER1」という名称のサブディレクトリが存在しさらに「98120100.MSV」と言うファイルのエントリがその下に存在していることを示している。このエントリには「98120100.MSV」と言うファイルの先頭のクラスタを示す「2」が書かれている。

【0119】FAT領域においては、「2」というFAT上のアドレスには「3」というアドレスが書かれており、更にアドレス「3」には「5」というアドレスを示す値が書かれている。このようにして離散的に書かれているデータをFATファイルシステム上でおっていくことで「98120100.MSV」というファイルは、クラスタ2→クラスタ3→クラスタ5→クラスタ6→クラスタ7→クラスタ8→クラスタ10→クラスタ11の順に板状メモリ1に記録されていることがわかる。これは図26のデータ領域内に示しているように、「98120100.MSV」というファイルが3つの離散的なファイル「98120100-1」「98120100-2」「98120100-3」で構成されることになる。なお上述した図22の説明から理解されるように、ファイル名はクラスタ200に記録されている。

【0120】このようなディレクトリ構造を持った板状メモリ1に新たにデータを書こうとした場合、図16のステップF 101において仮にルートディレクトリに特別なエントリファイルが作成されるわけであるが、これにより図24に示すように例えば「temptemp.tmp」と言うファイル名のファイルが作成される。ただし、この段階ではファイルエントリが作成されるだけのため、ファイルサイズは0とされている。図27は、この段階でのFATファイルシステム上でのデータの状態を示している。ルートディレクトリには「temptemp.tmp」という名称のサイズ0のファイルが作成されている。

【0121】この後、クラスタ1ではない一番若いクラスタ番号を持った空のクラスタを検索する。この検索は、FAT領域を順に見ていき「0」の書かれているアドレスを発見することで行われる。この例の場合はFATのアドレス「4」に対応するクラスタが空であるということがわかる。このアドレス「4」が図16のステップF 102 (図31) に示した高速ファイルオープン処理のCL (x) を求める処理になる。またその次に若いクラスタ番号を持った空のクラスタを検索するが、この場合はFATのアドレス「9」に相当するクラスタであ

りステップF102の高速ファイルオープン処理のCL(y)が「9」とされることがわかる。この後それぞれのクラスタ「4」、クラスタ「9」が実際のフラッシュメモリ上で対応するブロック番号を論理物理アドレス変換テーブルをもって求め、求められた物理アドレスにしたがって対応するブロックの消去を行うことになる。これによって図16のステップF102に示した高速ファイルオープン処理が完了したことになる。

【0122】図28はストリームデータを板状メモリ1に書き込んでいる途中の状態を表している。図28ではクラスタ4とクラスタ9にボイスデータが記録されている。なお、例えばこれらを「temptemp-1」「temptemp-2」とするが、これらは説明の都合上ふった名称であって実際のファイル管理システム上に管理されているものではない。

【0123】図32のような従来のデータ書き込み手順の場合、FATファイルシステムを利用して書き込みを行うため、この時点で例えばFAT領域のアドレス「4」にはクラスタ9にリンクする情報として「9」が書かれることになるが、本実施の形態ではそのような処理はこの段階では行わないため、FATのアドレス「4」には 図28のように「0」が書かれた状態となっている。

【0124】このときには、次に書込が行われるクラスタ12は既に消去がされている状態であり、クラスタ12が次に書き込むクラスタとして変数CL(x)にセットされ、更にその次に書き込みされるクラスタとしてクラスタ13が変数CL(y)にセットされてクラスタCL(y)のデータは消去されることになる。

【0125】図29は、入力されるストリームデータが終了した状態を表している。この場合、最終的に入力されたストリームデータは3クラスタのデータ量であったことになる。ここで新たに書き込まれた3番目のクラスタのデータを仮に「temptemp-3」とするが、これはファイル管理システムで管理された名称でないのは上記「temptemp-1」や「temptemp-2」と同様である。

【0126】続いて、これまでに書き込まれたストリームデータを実体のあるファイルとしてファイルシステムに登録する処理が必要になるが、それを図30を用いて説明する。つまり図16のステップF105のFAT更新であるが、ここでFATを走査して空のクラスタを示す「0」が書かれた一番小さいFAT上のアドレスを探しだす。この例(図29の時点)ではクラスタ4に対応するアドレス「4」の部分に「0」が書かれていることが検出される。つまりクラスタ4が新たに記録したストリームデータの先頭になるわけである。さらに次の空のクラスタを示すFAT上のアドレスを探す。この例(図29)の場合クラスタ9であることがわかるので、先程のクラスタ4を示すFATのクラスタ4に相当する位置

(アドレス「4」)に次ぎに再生されるデータがクラスタ9であることを示す「9」を書き込む(図30参照)。

【0127】次にクラスタ9のデータを読みだしてみる。このときデータの書き込みがあると判断されるので更に次に空のクラスタで若いクラスタ番号のクラスタを、同様にFAT上で探す。この例の場合クラスタ12であることがわかり、またクラスタ12は空ではないので、FATのクラスタ9を示す位置に次ぎに再生されるクラスタがクラスタ12であることを示す「12」を書き込む(図30参照)。

【0128】さらに次に若いクラスタ番号のクラスタをFATにおいて探すとクラスタ13であることがわかるが、実際にクラスタ13にはストリームデータは書き込まれておらず消去された状態にあることが、クラスタ13を読み出すことによってわかる。そこでFATのクラスタ12を示す位置にはこのクラスタ12が新たに記録されたストリームデータの最後のクラスタであることを示す「FFF」を書き込む(図30参照)。

【0129】これによって図30のように、FATとされている領域の4番目には「9」、9番目には「12」、12番目には「FFF」が書かれて、離散的に書き込まれたストリームデータのリンクが形成されたことになる。

【0130】次にディレクトリエントリの更新を行うが、これはファイル名「temptemp.tmp」としたファイルサイズ0のファイルのファイルサイズを、ストリームデータの書き込みに応じた値にして、さらにサブディレクトリの領域に例えば図30の例であればファイル名「temptemp.tmp」を「98120200.MSV」と名称を変更して記録するものである。このとき「98120200.MSV」の最初のクラスタ番号が4であることも一緒に更新する。つまりファイルサイズの更新とディレクトリ変更を伴うファイル名変更を行うものである。さらに、クラスタ201にはファイル名称が「98120200.MSV」であることがファイルシステムによって認識されるように「98120200.MSV」を書き込む。

【0131】この処理によって図16のステップF101において行った高速処理マーキングとしての「temptemp.tmp」と言う名称のファイルの消去も行われたことになる。

【0132】この状態を図25に示している。上記図24の状態においてはルートディレクトリの直下にVOICEと言う名称のサブディレクトリと同列に存在していた「temptemp.tmp」という名称のファイルは、ディレクトリの変更を伴う名称変更をした結果、図25に示したようにルートディレクトリの下に「VOICE」という名称のサブディレクトリの下に「FOLDER1」という名称のサブディレクトリの下に、「98

120100. MSV」という名称のデータと同列に新たに記録されたストリームデータとして、「98120200. MSV」という名称でファイルシステムに認識されるようになる。これによってルートディレクトリの直下に存在していた「temptemp. tmp」という名称のファイルはファイルシステムから消去されたことになる。

【0133】以上の処理を終えたら、図16のステップF106のファイルクローズを行い、一連の処理が完了する。

【0134】なお本例では、新たに記録されたストリームデータのファイル名を「98120200. MSV」としているが、このようなファイル名は装置が適当に割り振っても良いし、また記録開始前または記録終了時などにユーザーに入力させるようにしても良い。

【0135】以上の説明から理解されるように、本例では高速書込処理として、ストリームデータは連続的にクラスタ単位で書込を行っていくようにし、FATの更新はストリームデータの書込終了後に実行される。また上述したように、本例のアドレス変換テーブルでは、未使用の論理アドレスと物理アドレスについても、予め対応関係が記録されているものとなっている。

【0136】このため、ストリームデータを連続して記録している期間には、FATやアドレス変換テーブルの更新は行なわれないことになる。従ってストリームデータの記録に関して、単位時間当たりの書込可能なデータ量を、図32で説明した従来の処理に比べて大幅に増大できる。つまり単位時間当たりの書込可能なデータ量は、ほぼハードウェアのスペックによる上限まで増大させることが可能となる。これにより、高ビットレートのストリームデータに対しても対応可能となる。例えばボイスデータ、高音質オーディオデータはもとより、データ量の多い動画データなどについても、十分にリアルタイムな記録動作が可能となる。さらに、実質的にはデータ書込速度が大幅に高速化されることにもなり、またFATの更新のためのアクセスは1回のみとなり、大幅に減少することになるため、記録動作にかかる消費電力も著しく低減できる。

【0137】また、高ビットレートのストリームデータのリアルタイムな書込に対応できることは、大容量のバッファメモリを設けたり、或いは他の機器から出力されるオーディオデータや動画データなどを記録する際に、そのストリームデータの供給機器側からの出力タイミングを制御するなどのシステム処理を行うことなどが不要となることを意味し、ドライブ装置としてのハードウェア及びソフトウェアの両面での構成の簡略化を実現できることにもなる。

【0138】6. 復活処理
ところで上記高速書込処理により、例えば高ビットレートのストリームデータでもリアルタイムな書込が可能と

なるが、上述した処理の説明からもわかるように、FAT更新はストリームデータの書込完了後において行われるものであり、またFAT更新が完了して初めて書き込まれたストリームデータが有効化、つまり再生可能なデータとされるものである。これは、もしストリームデータの書込途中もしくは書込が完了した時点などにおいて、FAT更新が行われる前に例えば停電やその他の障害により電源遮断が発生し、FAT更新が不能となってしまった場合は、書き込んだストリームデータが無効なファイル、つまりFATで管理されていないため再生不能なファイルとなってしまふ。

【0139】これが例えば他の記録媒体からダビングしたデータである場合などは、再度ダビングをすることにより事なきを得るが、マイクロフォン103から集音した音声を記録していた場合や、或いは撮像機器で撮像された動画データをリアルタイムで記録していた場合、さらには例えば放送受信機器で受信復調された放送音声を記録していた場合など、再度そのストリームデータを取り込むことができない場合は、大きな問題となってしまふ。このため本例では、一旦板状メモリ1に書き込んだデータについては、もしFAT更新が不能となった場合でも、復活処理により有効なデータに復活させることができるようにしている。

【0140】この復活処理としてマイクロコンピュータ109が実行する処理を図21に示すが、この復活処理を実現するために、書込時には上述した図17の処理が行われているものである。つまり、上述したようにクラスタ単位での書込動作の際には、書込規則として、各時点で空きクラスタのうちで最も若いクラスタナンバ（論理アドレス）のブロックにストリームデータの書込を行なうようにしていることと、次及びその次に若い2つのクラスタナンバのブロックについてデータ消去を行うようにしていることである。

【0141】このような処理を前提とすることで、図21の復活処理が可能となる。復活処理の際には、まずステップF201として、FATで管理されている未使用クラスタ（＝空きクラスタ）のうちで、その時点で最も若いクラスタナンバ（論理アドレス）を変数CL(x)にセットする。なお、上記図18、図19、図20で説明したような書込動作について復活処理が実行されるのは、例えば図18のようにファイルF2が板状メモリ1に記録されはしたが、FATが依然として未更新の図19のままである場合である。従って、このステップF201で最も若いクラスタナンバと判断されるクラスタは、実際にはファイルF2の最初のクラスタが書き込まれたクラスタナンバとなる。つまり図18、図19の場合は、クラスタCL(5)となる。

【0142】次にステップF202で、クラスタナンバCL(x)となるクラスタに対してデータ読み出しを実行し、ステップF203で、実データが存在するか否

か、つまり、データ消去が実行されているブロックであるかを判別する。上記例の場合、クラスタCL (5) にはデータが記録されているため、ステップF204に進むことになる。

【0143】ステップF204では、FAT (図19の状態) において空きクラスタとされるうちで、次に若いクラスタナンバを変数CL (x) にセットする。そしてステップF202、F203の処理を行う。つまり同様にそのクラスタナンバCL (x) のブロックの読み込み及びデータ存在の確認を行う。上記例の場合、クラスタCL (5) の次に若いクラスタナンバであるクラスタCL (7) について読込及びデータ確認が行われることになる。そしてクラスタCL (7) にはデータが記録されているため、再びステップF204に進む。

【0144】このようなステップF201～F204の処理により、データが記録されている最後のブロックが検出されることになる。つまり、例えば上記例の場合は、空きクラスタについてナンバが若い順に、クラスタCL (5)、CL (7)、CL (A)、CL (B)、CL (C)、CL (D)、CL (E) とそれぞれ読出及びデータ存在の確認が行われる。ここでクラスタCL (E) まではデータ存在が確認され、ステップF204に進むことになるが、そのステップF204では、変数CL (x) としてクラスタCL (F) がセットされ、ステップF202、F203で読出及びデータ存在確認が行われることになる。このとき、上述した書込処理において、ストリームデータを書き込んだ次及びその次のブロックに対しては消去を実行しているため、データ無しと判断されることになる。

【0145】従って、これによってクラスタCL (5)、CL (7)、CL (A)、CL (B)、CL (C)、CL (D)、CL (E) に有効化すべきデータが存在するということが確認でき、ステップF205では、現在のクラスタCL (x) (=CL (F)) が異常終了にかかるクラスタであると判断する。そしてステップF206、F207では、有効化すべきと判断されたクラスタCL (5)、CL (7)、CL (A)、CL (B)、CL (C)、CL (D)、CL (E) がFAT上でリンクされるようにし、またこれが有効なファイルとされるようにディレクトリエントリを更新する。これによりFATは図20の状態に更新され、つまりファイルF2が有効なファイルとして復活されることになる。

【0146】このように復活処理が行われることで、たとえ上述した高速書込処理として、FAT更新がストリームデータ記録後に一括して実行されるようにしても、電源遮断等の事故の場合も対応できることになる。つまり上記高速書込処理により、書込済のストリームデータに対してFAT未更新状態が発生することがあり得ることとなるが、この復活処理が実行できることで、FAT未更新状態が発生してもそれを解消できることとなり、

問題ないものとなる。

【0147】なお、ファイルF2としての全ストリームデータの書込が完了した後にFAT更新不能となった場合は、このように復活処理によりそのファイルF2を完全に復活させることができるが、ストリームデータの書込途中において電源遮断などで処理が中断されてしまった場合は、その書込が行われた部分までが復活処理の対象となることはいうまでもない。例えば図18のファイルF2についてクラスタCL (5)、CL (7)、CL (A)、CL (B) まで書き込んだ時点で動作不能となった場合は、ステップF201～F204の処理で、クラスタCL (C) についてデータ無し (消去状態) と判断されることになるため、ステップF206、F207の更新処理では、クラスタCL (5)、CL (7)、CL (A)、CL (B) がリンクされた1つのファイルとなるように更新される。

【0148】なお、以上の復活処理は、ユーザー操作により開始されるようにしても良いし、マイクロコンピュータ109がFAT未更新状態を自動検出して実行しても良い。例えばマイクロコンピュータ109は、電源が回復した時点で、板状メモリ1において高速処理マーキングがオフされていないなど適正なファイルクローズが行われていないことを検出したら、即座に復活処理を実行するようにすれば、ユーザーに対してはFAT更新状況によるファイルの有効/無効を意識させることなく、あくまでも書込を行った部分については有効であるように認識させることができる。また高速マーキング処理において、上述した「temptemp.tmp」のような暫定的なファイル名を設定し、最終的にそれを消去して正規のファイル名を設定する場合は、電源回復時点などで「temptemp.tmp」というファイル名が検出された場合は、適正なファイルクローズが行われていないと判断することができる。従ってそのような場合に上記復活処理を開始すればよい。

【0149】以上、実施の形態について述べてきたが、本発明はこれらの構成及び動作に限定されるものではない。特に高速書込処理や復活処理の細かい手順としては各種の変形例が考えられる。

【0150】例えば高速書込処理において最後に書き込みされたクラスタの判断として、消去済みのクラスタを検出したか否かという方法をとっているが、ストリームデータの書き込みが終了した時点で記録の最後のクラスタ番号をマイクロコンピュータ109が記憶しておいて、クラスタのリンクを作成する段階でマイクロコンピュータ109が記憶した最後のクラスタ番号と照合することでリンクの最後のクラスタを容易に判別することが可能である。

【0151】また、記録終了後に記録したクラスタのリンクをさせていく処理を簡略化する目的でストリームデータの記録開始位置を書き込み処理中であることを表す

「temptemp.tmp」の関連情報の一部として、図27に示したサブディレクトリに書かれた「98120100.MSV」の開始位置として記憶しているクラスタアドレス「2」と同様に、クラスタアドレス「4」を記録するようにしても良い。このようにすることでリンク情報作成時においてFAT領域を参照して記録開始位置を確認することが不要となり、またFAT領域を参照するのであれば記録開始位置が正しいことを知ることでもある。

【0152】さらに、記録終了後に記録したクラスタをリンクをさせていく処理を簡略化する目的で、ストリームデータを板状メモリ1に記録するたびに、たとえばドライブ装置内のRAM111にクラスタ番号のみ記録順序が判別可能なように記録しておき、記録されたクラスタのリンクの処理時にはRAM111に記憶しておいたクラスタ番号に基づいてリンクの作成を行っていけば、板状メモリ1上に形成されているFATの読み出しや消去済みとされているクラスタにデータが書き込まれているかどうかを判断する処理は省略可能となる。このようにすることによって正常に記録が終了してFATの更新処理などが行われる場合には、新たに記録されたストリームデータに対するクラスタのリンク処理がより高速に行えることになる。

【0153】また、本発明のシステムとしては、図1のような板状メモリに限定されるものではなく、他の外形形状とされた固体メモリ媒体（メモリチップ、メモリカード、メモリモジュール等）でも構わない。また、これまで説明したファイルシステムのフォーマットも、例えば実際に応じてその細部の規定などは変更されて構わない。更には、フラッシュメモリ容量のバリエーションも図9に示したものに限定されるものではない。もちろん、本発明の記録媒体のメモリ素子はフラッシュメモリに限られず、他の種のメモリ素子でもよい。

【0154】

【発明の効果】以上の説明からわかるように、本発明の記録方法、記録装置では、ストリームデータ等の主情報を連続して記録している期間には、ディレクトリ情報やリンク情報等の管理情報の更新は行なわれないことになり、従って主情報の記録に関して、単位時間当たりの書込可能なデータ量を大幅に増大できる。つまり単位時間当たりの書込可能なデータ量は、ほぼハードウェアのスペックによる上限まで増大させることが可能となる。これにより、より高ビットレートのストリームデータに対しても対応可能となるという効果がある。さらに、実質的にはデータ書込速度が大幅に高速化されることになるという効果もあり、また管理情報やアドレス変換情報の更新のためのアクセス回数の大幅減少から記録動作時の消費電力も著しく低減できる。

【0155】また本発明の管理方法によれば、主情報の記録動作の中断又は完了の後に、管理情報更新動作が実

行不能となった場合には、その後の動作時において、主情報の記録に用いられたブロックを判別し、その判別に応じて管理情報を更新して、更新した管理情報を記録媒体に記録することで、主情報の記録動作においてその中断又は完了までに記録された主情報を有効化させる記録データ復活動作が可能となる。従って、本発明の記録方法、記録装置のように管理情報の更新が主情報の書込完了後に行われるシステムであって、例えば突然の電源遮断などによって管理情報更新が不能となった場合でも、記録済データについては保護できることになる。これによりシステムの信頼性を向上させることができる。

【図面の簡単な説明】

【図1】本発明の実施の形態の板状メモリの外形形状を示す平面図、正面図、側面図、底面図である。

【図2】実施の形態のファイルシステム処理階層の説明図である。

【図3】実施の形態の板状メモリの物理的データ構造の説明図である。

【図4】実施の形態の板状メモリの管理フラグ内容の説明図である。

【図5】実施の形態の板状メモリにおけるデータ更新処理と物理アドレス及び論理アドレスの概念の説明図である。

【図6】実施の形態の論理—物理アドレス変換テーブルの管理形態の説明図である。

【図7】実施の形態の論理—物理アドレス変換テーブルの構造の説明図である。

【図8】論理—物理アドレス変換テーブルにおける未使用ブロックの管理形態を説明するための説明図である。

【図9】実施の形態の板状メモリのフラッシュメモリ容量／ブロック数／1ブロックの容量／1ページの容量／論理—物理アドレス変換テーブルのサイズの関係の説明図である。

【図10】実施の形態のドライブ装置のブロック図である。

【図11】実施の形態における板状メモリとセット本体のマイクロコンピュータとのインターフェイスを概念的に示す説明図である。

【図12】FAT構造の説明図である。

【図13】FATによるクラスタ管理形態の説明図である。

【図14】ディレクトリの内容の説明図である。

【図15】サブディレクトリ及びファイルの格納形態の説明図である。

【図16】実施の形態の高速書込処理のフローチャートである。

【図17】実施の形態の高速書込処理のフローチャートである。

【図18】実施の形態の高速書込動作例の説明図である。

【図19】実施の形態の高速書込処理前のFATの説明図である。

【図20】実施の形態の高速書込処理後のFATの説明図である。

【図21】実施の形態の復活処理のフローチャートである。

【図22】実施の形態のFATによる管理構造の説明図である。

【図23】実施の形態のデータ書込前のディレクトリ構成例の説明図である。

【図24】実施の形態のデータ書込中のディレクトリ構成例の説明図である。

【図25】実施の形態のデータ書込後のディレクトリ構成例の説明図である。

【図26】実施の形態のデータ書込前のデータ状態の説明図である。

【図27】実施の形態のデータ書込中のデータ状態の説明図である。

明図である。

【図28】実施の形態のデータ書込中のデータ状態の説明図である。

【図29】実施の形態のFAT更新前のデータ状態の説明図である。

【図30】実施の形態のFAT更新後のデータ状態の説明図である。

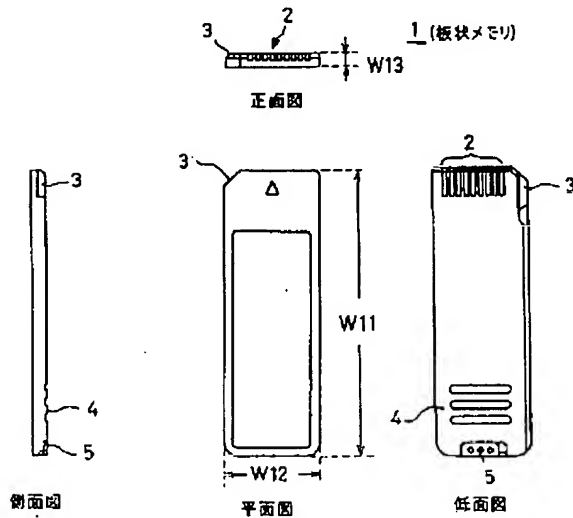
【図31】実施の形態の高速書込処理のフローチャートである。

【図32】従来の書込処理のフローチャートである。

【符号の説明】

1 板状メモリ、2 端子部、4 凹凸部、5 スライドスイッチ、100セット本体、103 マイクロフォン、104 マイクアンプ、105 スピーカアンプ、106 スピーカ、107 表示ドライバ、108 表示部、109マイクロコンピュータ、112 操作部、120 着脱機構

【図1】

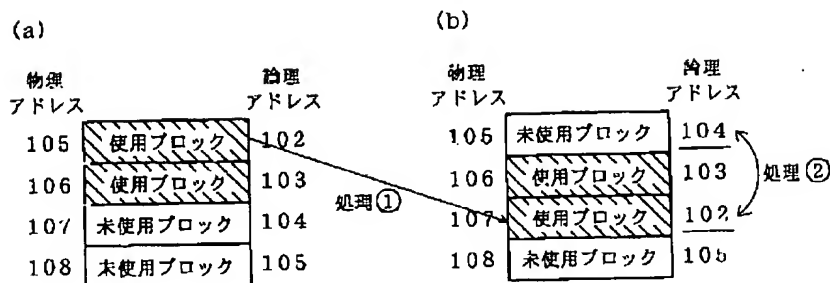


【図2】

アプリケーション処理
ファイル管理処理
論理アドレス管理
物理アドレス管理
フラッシュメモリアクセス

ファイルシステム処理階層

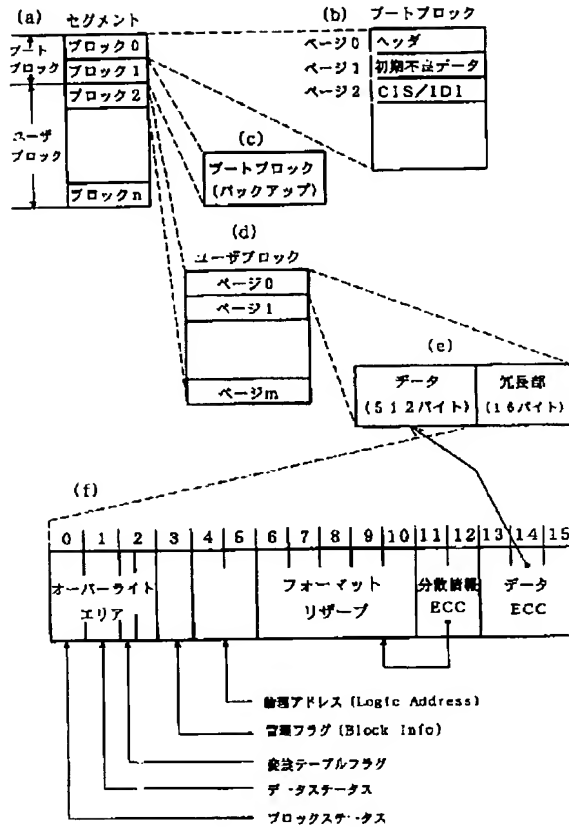
【図5】



データ更新処理前

データ更新処理後

【図3】

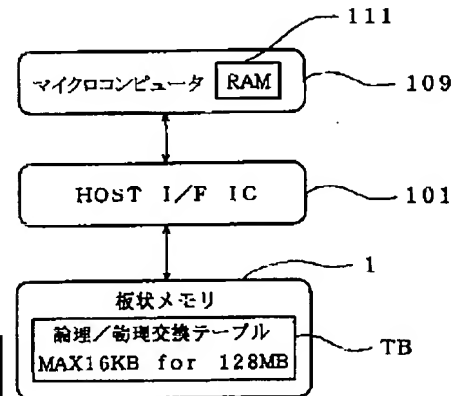


【図4】

管理フラグ

ビット	定義
7	リザーブ
6	リザーブ
5	アクセス許可 (1:free 0:Read Protected)
4	コピー禁止指定 (1:OK 0:NG)
3	変換テーブルフラグ (1:無効 0:テーブルブロック) *最終セグメントのみ有効
2	システムフラグ (1:ユーザーブロック 0:ブートブロック)
1	リザーブ
0	リザーブ

【図11】



【図6】

物理アドレス (2Byte)	論理アドレス (2Byte)
物理アドレス 0x0003	←論理アドレス 0x0000
物理アドレス 0x0004	←論理アドレス 0x0001
物理アドレス 0x0004	←論理アドレス 0x0002
物理アドレス 0x0005	←論理アドレス 0x0003
物理アドレス 0x01A8	←論理アドレス 0x0004
物理アドレス 0x0006	←論理アドレス 0x0005

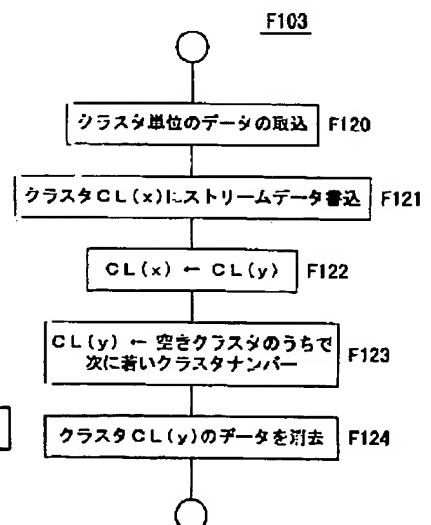
論理アドレス順に、対応する物理アドレスを格納する。

【図14】

ファイル名1本分のディレクトリの構造。〈 〉はバイト数。

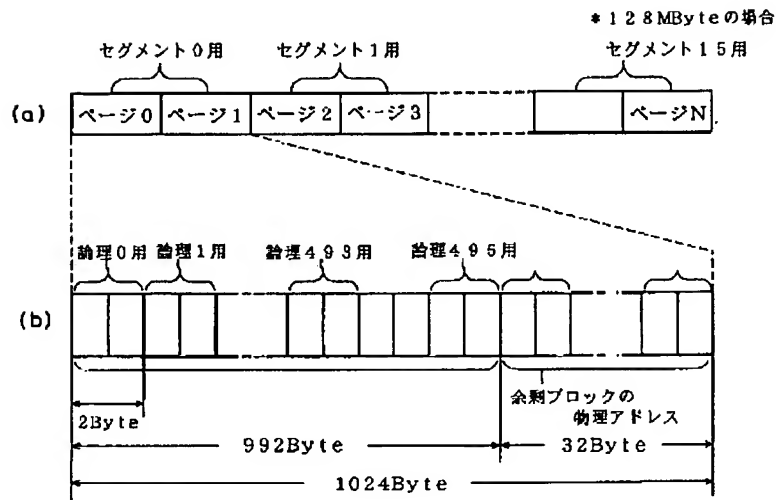
ファイル名(8)	拡張子(3)	属性(1)	リザーブ(10)	時刻(2)	日付(2)	先頭クラス(2)	サイズ(4)
----------	--------	-------	----------	-------	-------	----------	--------

【図17】

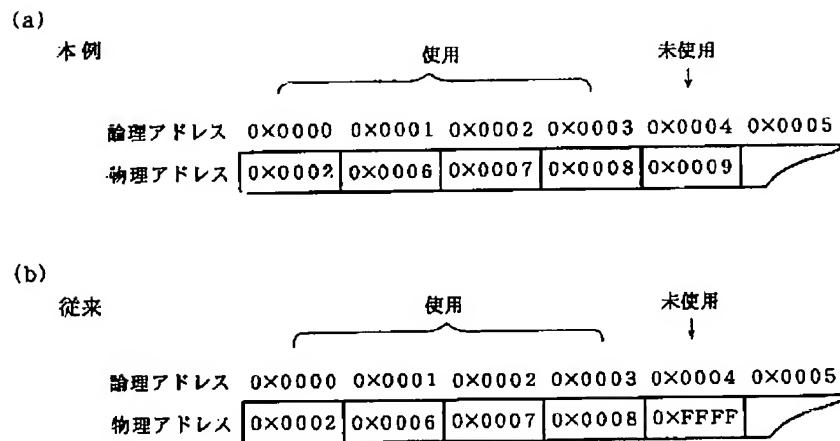


【図7】

論理／物理アドレス変換テーブル



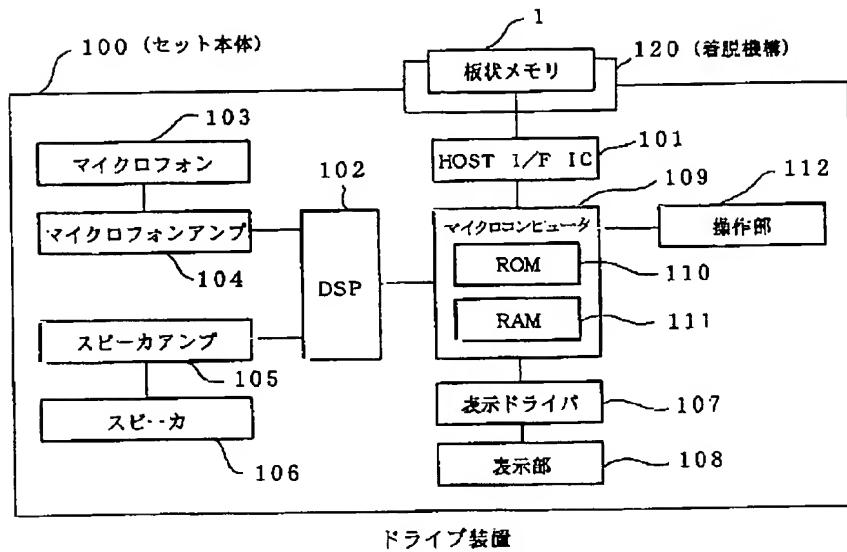
【図8】



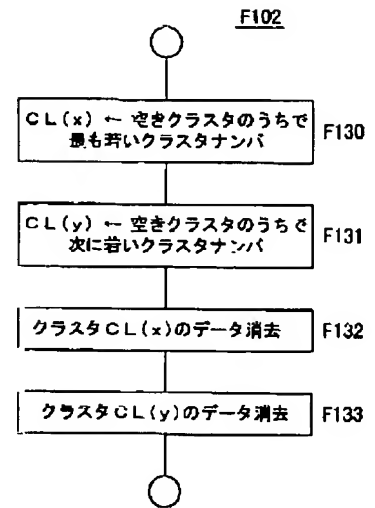
【図9】

フラッシュメモリ 容量	ブロック数	1ブロックの 容量	1ページの 容量	論理／物理アドレス 変換テーブルサイズ
4MB	512 (1セグメント)	8KB (16page)	(512+16) B	1KB (2page)
8MB	1024 (2セグメント)	8KB (16page)	(512+16) B	2KB (4page)
16MB	2048 (4セグメント)	8KB (16page)	(512+16) B	4KB (8page)
	1024 (2セグメント)	16KB (32page)	(512+16) B	2KB (4page)
32MB	2048 (4セグメント)	16KB (32page)	(512+16) B	4KB (8page)
64MB	4096 (8セグメント)	16KB (32page)	(512+16) B	8KB (16page)
128MB	8192 (16セグメント)	16KB (32page)	(512+16) B	16KB (32page)

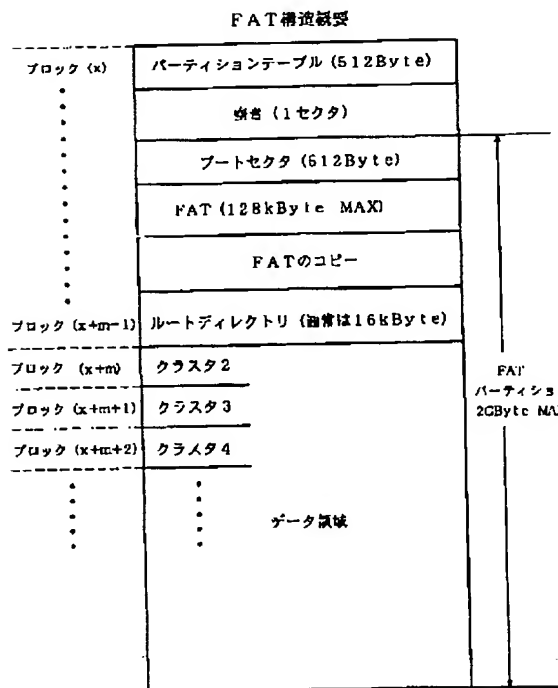
【図10】



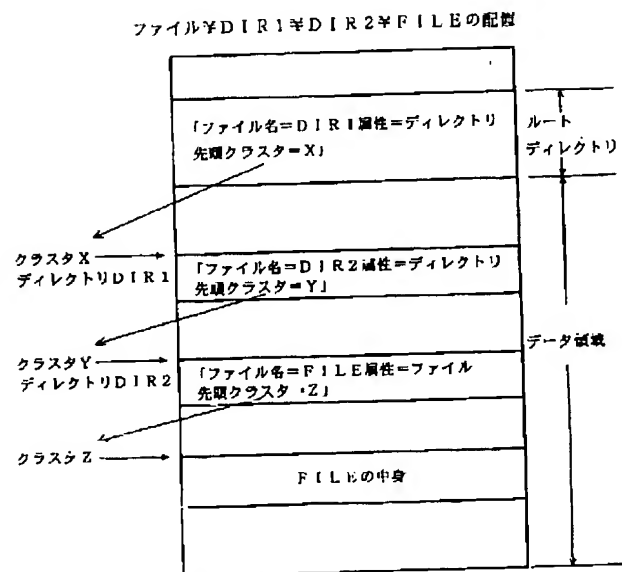
【図31】



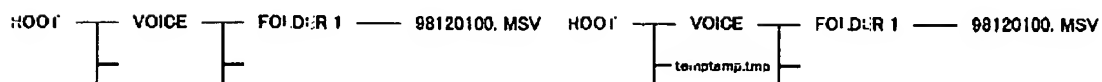
【図12】



【図15】



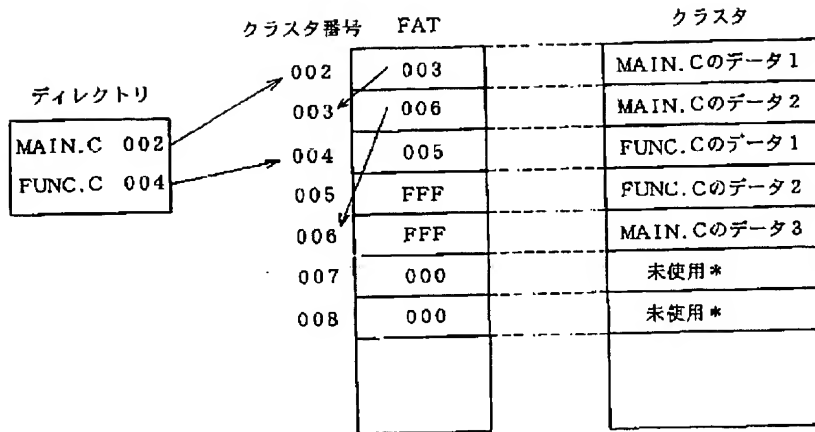
【図23】



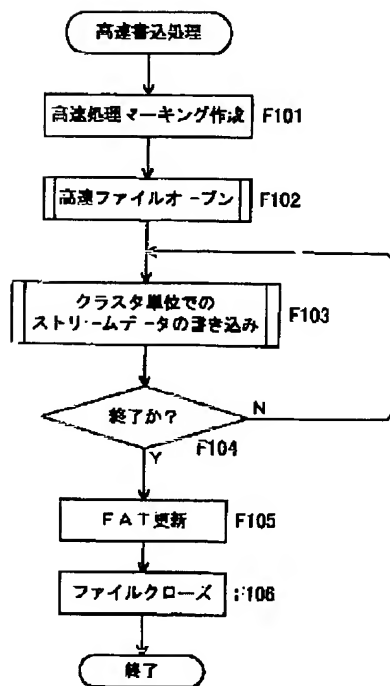
【図24】

【図13】

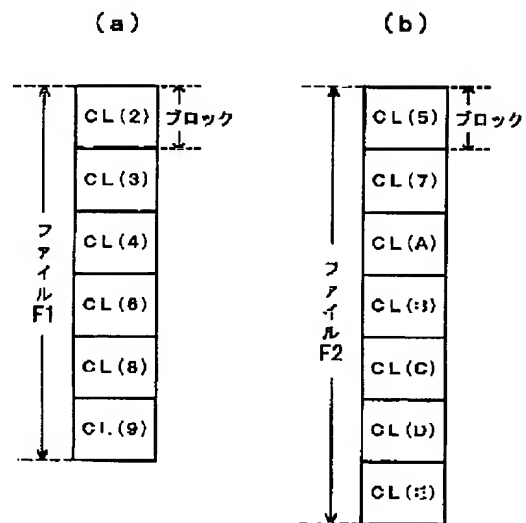
FAT概念図



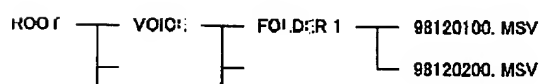
【図16】



【図18】



【図25】



【図19】

FAT (記録前)

* () 内は対応するクラス番号
数値は16進表記

(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
		003	004	006	000	008	000
(8)	(9)	(A)	(B)	(C)	(D)	(E)	(F)
009	FFF	000	000	000	000	000	000
(10)	(11)	(12)	(13)	(14)	(15)	(16)	(17)
000	000	000	000	000	000	000	000
(18)	(19)	(1A)	(1B)	(1C)	(1D)	(1E)	(1F)
000	000	000	000	000	000	000	000
(CLmax-7)	(CLmax-6)	(CLmax-5)	(CLmax-4)	(CLmax-3)	(CLmax-2)	(CLmax-1)	(CLmax)
000	000	000	000	000	000	000	000

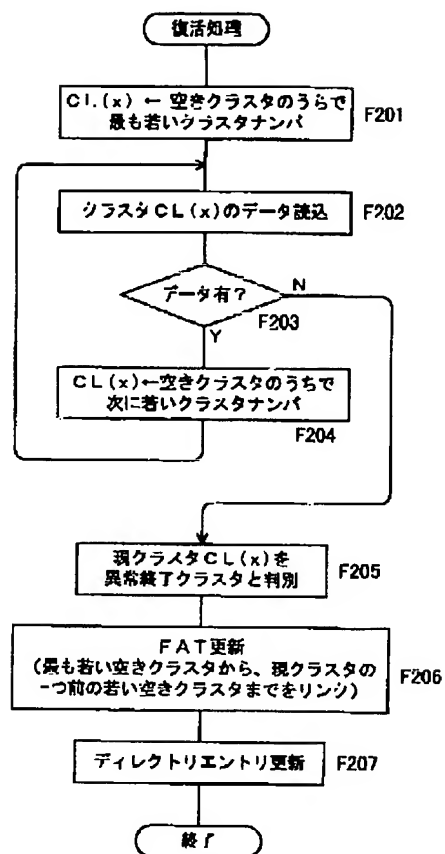
【図20】

FAT (記録後)

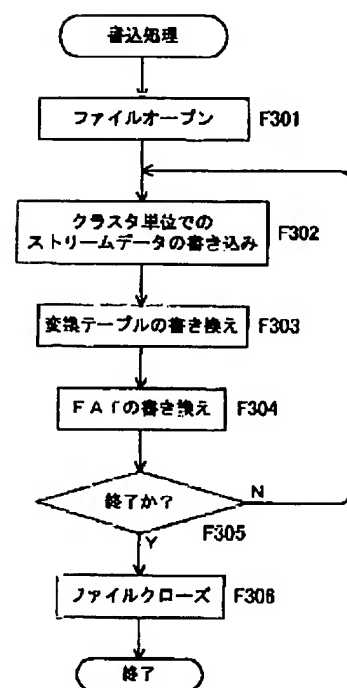
* () 内は対応するクラス番号
数値は16進表記

(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)
		003	004	006	007	008	00A
(8)	(9)	(A)	(B)	(C)	(D)	(E)	(F)
009	FFF	00B	00C	00D	00E	FFF	000
(10)	(11)	(12)	(13)	(14)	(15)	(16)	(17)
000	000	000	000	000	000	000	000
(18)	(19)	(1A)	(1B)	(1C)	(1D)	(1E)	(1F)
000	000	000	000	000	000	000	000
(CLmax-7)	(CLmax-6)	(CLmax-5)	(CLmax-4)	(CLmax-3)	(CLmax-2)	(CLmax-1)	(CLmax)
000	000	000	000	000	000	000	000

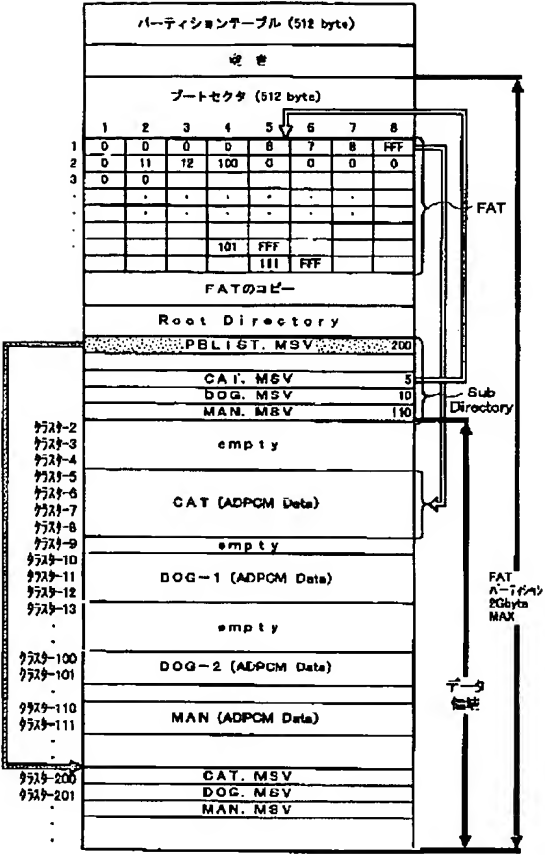
【図21】



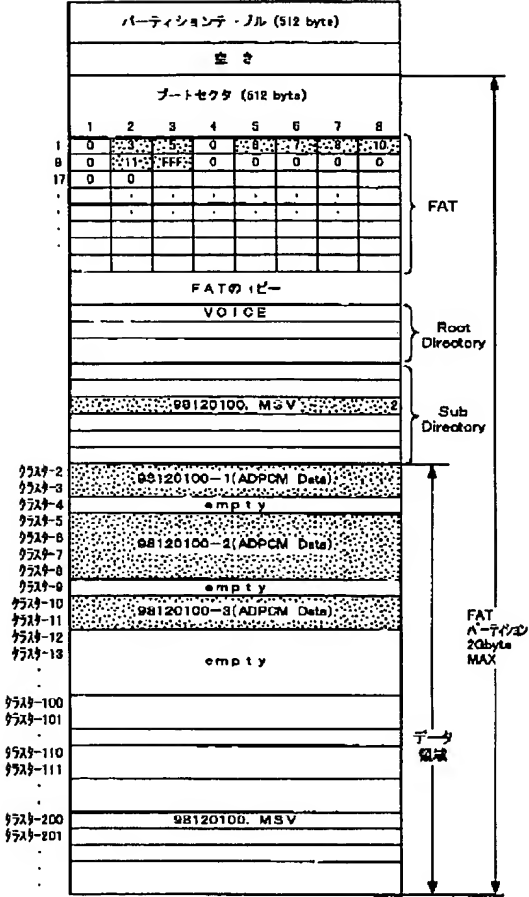
【図32】



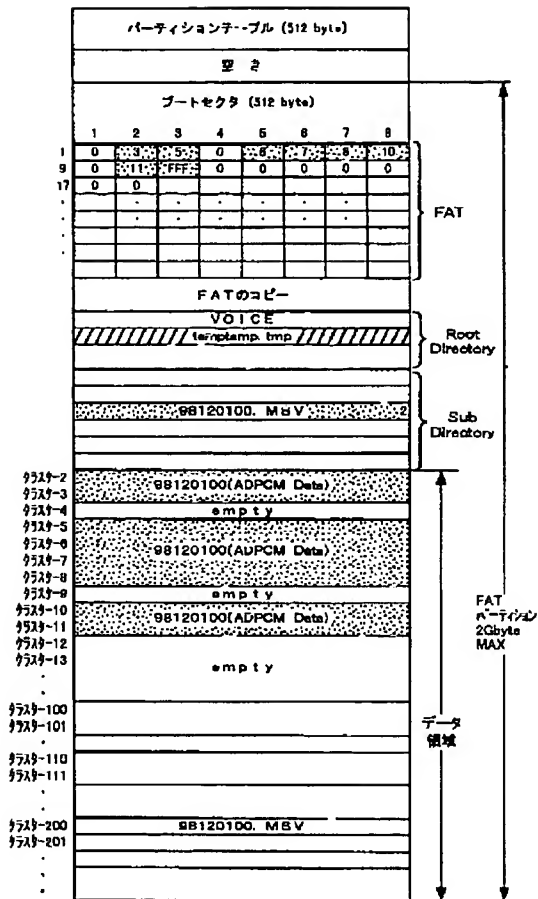
【図22】



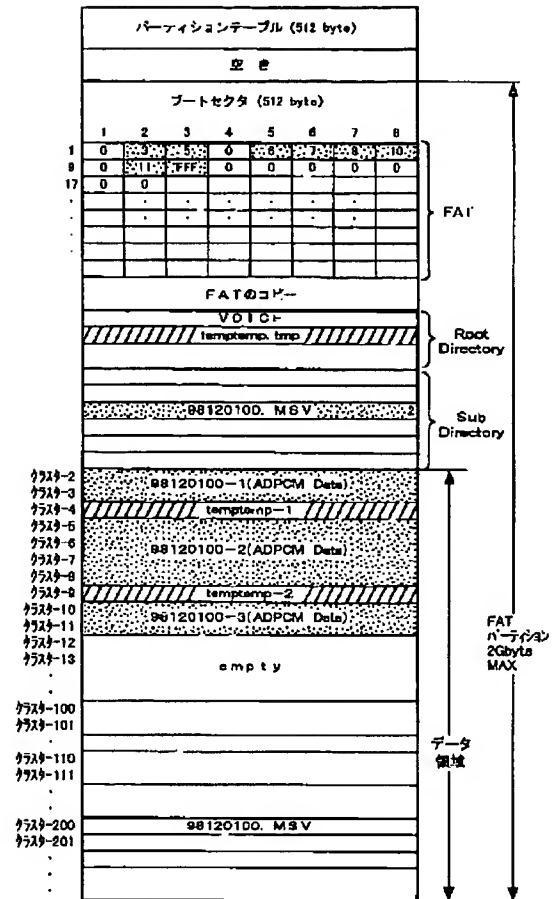
【図26】



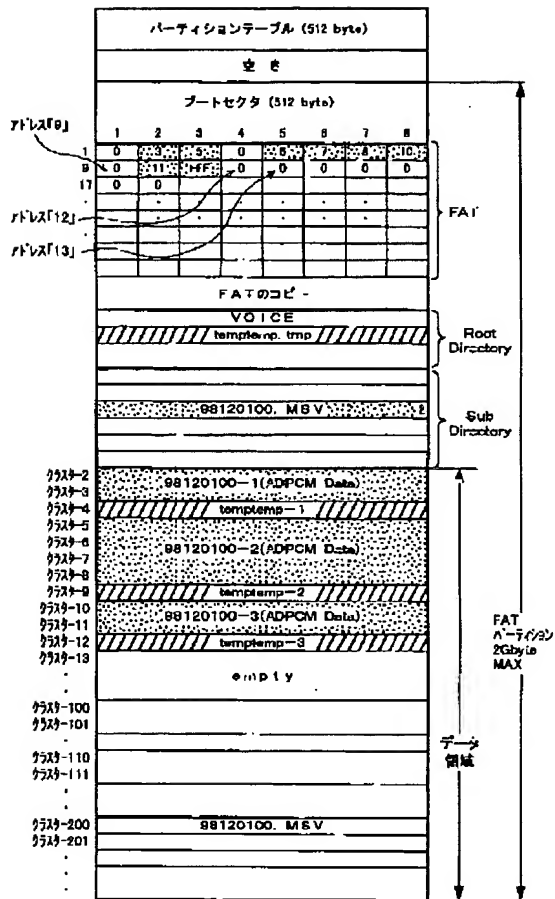
【図27】



【図28】



【図29】



【図30】

